Generation of Synthetic Image Sequences for Car Maneuvers Extracted from Image Sequence Evaluation

Master Thesis

by

V. Jeyakumar

Advisors: Prof. Dr. Amitabha Mukherjee (India)

Prof. Dr. H.- H. Nagel, Dr. M. Haag (Germany)

to the

DEPARTMENT OF MECHANICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

April, 1999

15.04.99

CERTIFICATE

It is certified that the work contained in the thesis entitled, **Generation of Synthetic Image Sequences for Car Manuevers Extracted from Image sequence Evaluation** by **Mr. V. Jeyakumar**, has been carried out under my supervision as well as under the supervision of Prof. Dr. H. H. Nagel and Dr. M. Haag, Department of Computer Science, University of Karlsruhe, Germany during V. Jeyakumar's visit to Germany from June '98 to Feb '99.

Dr. Amitabha Mukherjee,
Associate Professor,
Department of Mechanical Engg,
Indian Institute of Technology,
Kanpur - 208016.

April, 1999

Abstract

System that analyze video data generate voluminous output that is impossible to scrutinize manually. This work builds on past work that generates conceptual descriptions (abstractions) of visual events in the traffic domain. Based on this, synthetic image sequences are reconstructed for the car maneuvers. The parameterized vehicle model is projected onto a specific lane in a static image and is maneuvered by means of a motion model whose inputs are provided by decoding the conceptual description. Various cases from simple - i.e. a car on a straight road - to more complex maneuver sequences, for example lane changing, are to be analyzed. This provides an easy method for humans to verify the output of a video analysis. It is also of interest on its own as a reconstruction from abstract motion description, eg: in video search or in movie generation.

Contents

Chapter 1

Introduction

1.1 Motivation

For the last two decades, the *Evaluation of Image Sequences (EIS)* has progressed into an increasingly recognized sub-discipline of Artificial Intelligence (AI). The *EIS* expresses the behavior of an agent recorded by an image sequence in the form of *natural language texts*. Around the world, various researchers working in this field are following different types of approaches. [Nagel 88; Nagel 91] works are based on Logical Programming whereas the work of [Buxton & Gong 95] is based on *Bayesian* Techniques.

The fig 1.1 shows how *EIS* has been used to produce Natural Language description of traffic behavior at Karlsruhe. First of all X-track system detects the moving object from image sequences by using optical flow techniques (see section 2.1) and dose the tracking of road vehicles by implementing various approaches (refer section 2.2). F-limette, which is discussed more elaborately in section 2.3, generates Conceptual descriptions from the time stamped data obtained from X-Track system based on Fuzzy Metric Temporal Logic (FMTL). Finally, Natural Language Text is produced by Discourse Representation Structures (DRS) ([Gerber & Nagel 98]) which is the part of F-Limette.

The primary task of the project to be reported here is to reverse the work which is mentioned above, i.e. generating synthetic image sequences(SIS) from the text which is given as a conceptual description. Only few attempts have been reported so far on this. Carrying out such a work will lead to form a loop, i.e. Image Sequence - to - Conceptual Description - then to - Synthetic Image Sequence. One can thereby study experimentally how an artificial system which understands text in order to generate the synthetic image sequence compares to a human being who visualizes the scene.

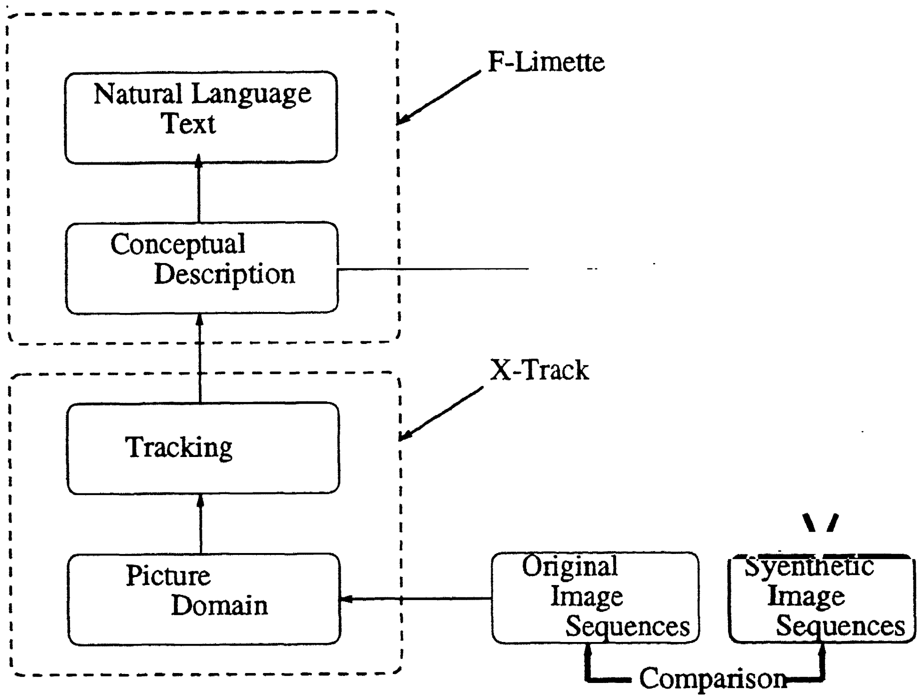


Figure 1.1: layered system structure indicating the stepwise transformation from Original video image sequences to Natural language text and transformation from Conceptual descriptions to Synthetic image sequences which is goal of this project.

Functionally also, this serves an important need. The sheer volume of analyzed data in any image analysis such as X-track is such that it cannot be scrutinized manually. Thus a inverse program that closes the loop is extremely valuable for checking the results of the visual analysis. In addition, scene reconstruction from abstract input is an important goal in itself in tasks such as video search for ex: find the scene where the red car catches up with the van.

1.2 Problem Definition

As we mentioned above, our aim is to generate Synthetic Image Sequences (SIS) from elementary conceptual descriptions which is the intermediate output between Tracking and Natural Language Text (see 1.1). Conceptual description as used in this work represent an abstraction for the concepts of speed and distance. Time is specified exactly, and space is inferred from the motion. In Natural Language Text both time and space may also be abstracted. The output of the driving agent will be actions such as steering, accelerating, or braking. The environment will consist of lane structures of

the road or intersection, other vehicles, traffic signals, pedestrians, and other structures such as buildings, posts, etc.

In this work, the environment will consist initially only of lane structures which will be given by geometric descriptions, and of other vehicles. According to the different complexity of potential maneuvers, we will analyze the following cases of car maneuvers:

- Single car on simple and straight road with varying velocity – Straight Lane (Chap. 4)
- Single car turning into an intersection scene – Turning (Chap. 5)
- Car following another car or sequence of cars – Car Following (Chap. 6)
- Car changing to a different lane – Lane Changing (Chap. 7)

The following models are needed to carry out the above experiments.

Conceptual Description: This is a set of fuzzy description for the car speed, its lane, its distance from other vehicles etc. Refer Section 4.2 for details.

Parameterized vehicle model: This is discussed briefly in Section 1.3. It will allow us to handle different types of vehicles. By changing parameters, we can model different families of vehicles.

Motion model: This will be required to maneuver the vehicles on the roads based on the input obtained from conceptual descriptions.

Finally, the synthetic image sequence(**SIS**) generated by our system will be superimposed on the original image sequences(**OIS**) recorded by a real static camera based on which the conceptual description has been generated. This will permit easy scrutinizing of the correctness of the conceptual description generated. However, dud to the inexact nature of the conceptual input it is not expected that there will be a very close match between the two.

1.3 Parameterized Vehicle Model

We are going to use a 3-D generic model to handle different ádmitted cars. The parameters are given in Fig. 1.2 and their descriptions are given in Table 1.1. Different types of vehicles are generated from this generic description by varying the 12 length parameters. Fig. 1.3 shows examples for five different specific vehicle models derived from the same generic model.

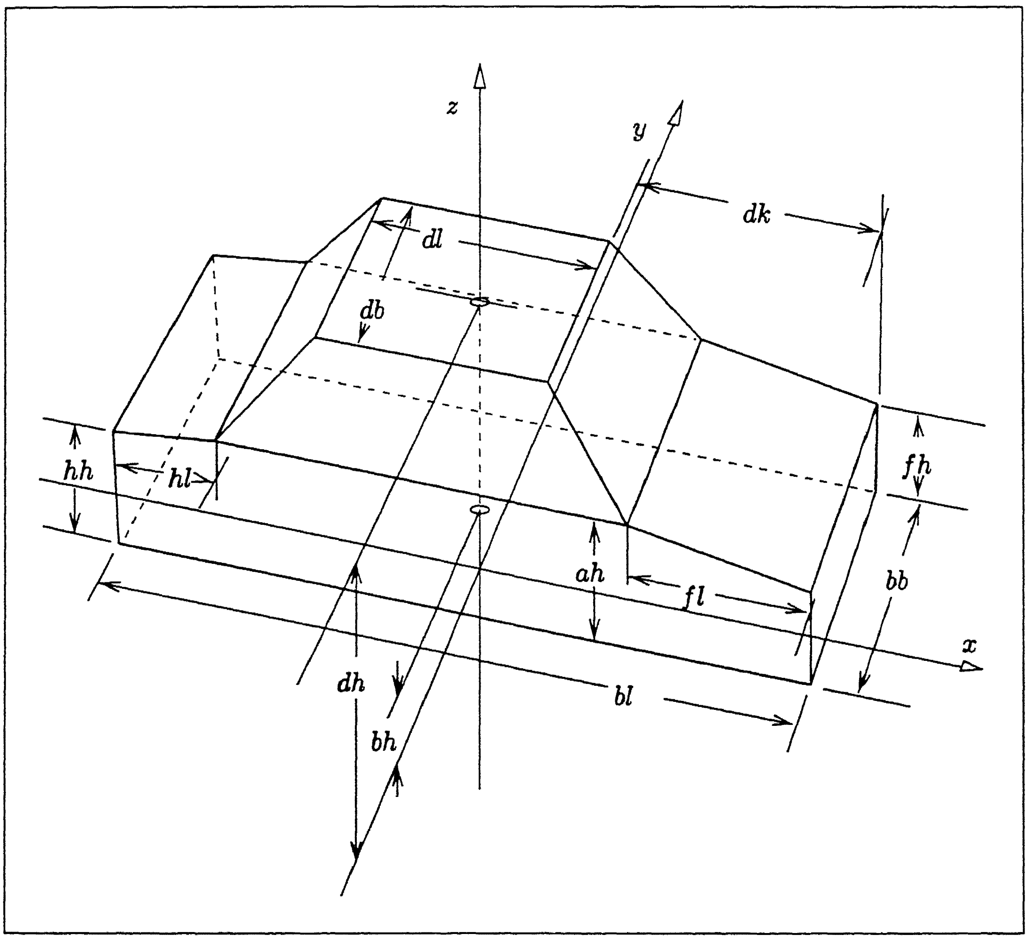


Figure 1.2: 3-D generic model with 12 length parameters (from [Koller et al. 93]).

Parameter	Abbrev.	Meaning
Bottom length	*bl*	Bottom length of the vehicle
Bottom width	*bb*	Bottom width of the vehicle
Bottom height	*bh*	Bottom height of the vehicle
Roof length	*dl*	Length of the vehicle's roof
Roof width	*db*	Width of the vehicle's roof
Roof height	*dh*	Height of the vehicle's roof over ground
Roof edge	*dk*	Distance between the front roof edge and the vehicle's front
Front length	*fl*	Length of the engine bonnet
Front height	*fh*	Hight of the vehicle's front
Rear length	*hl*	Rear length of the vehicle
Rear height	*hh*	Rear height of the vehicle
Bottom height to middle	*ah*	Bottom height of the vehicle at middle

Table 1.1: The abbreviations and the meaning of the 12 parameters (from ??).

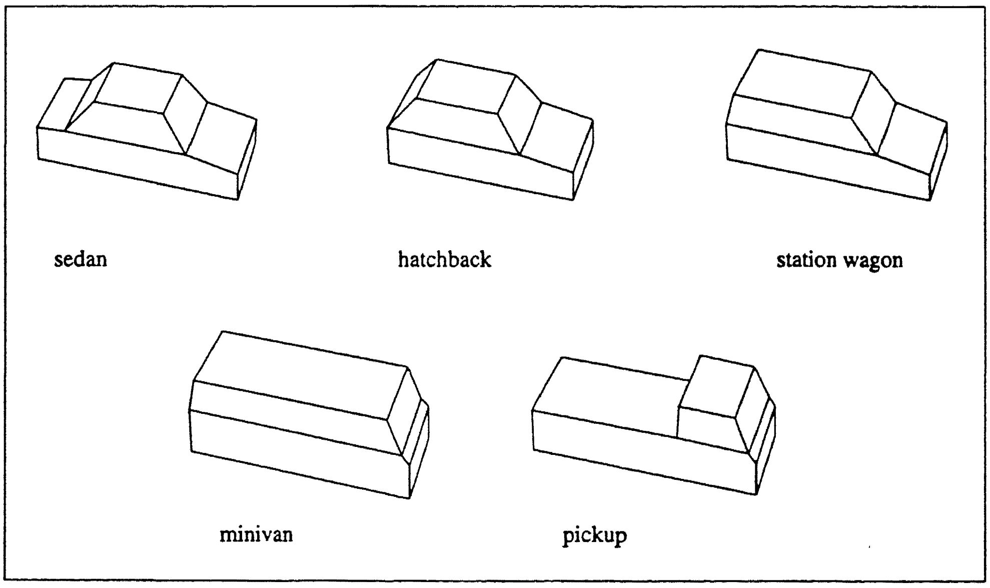


Figure 1.3: Examples of five different vehicle models derived from the same generic model (from [Koller et al. 93]).

1.4 Motion Model

We use a motion model that describes the dynamic behavior of a road vehicle without knowledge about the intention of the driver. Since we further assume that the motion is constrained to movements on a street plane, we get in the stationary case a simple circular motion with a constant magnitude of the translational velocity v and a constant angular velocity ω. The remaining three degrees of freedom of the external vehicle model instantiation are described by the position $\mathbf{p}(t) = (p_x(t), p_y(t), 0)^T$ of the vehicle center on the street plane and the orientation angle ϕ about the normal to the plane (the z-axis) through the vehicle center, i.e. the orientation of the principal axis of the vehicle model with respect to the scene coordinate system.

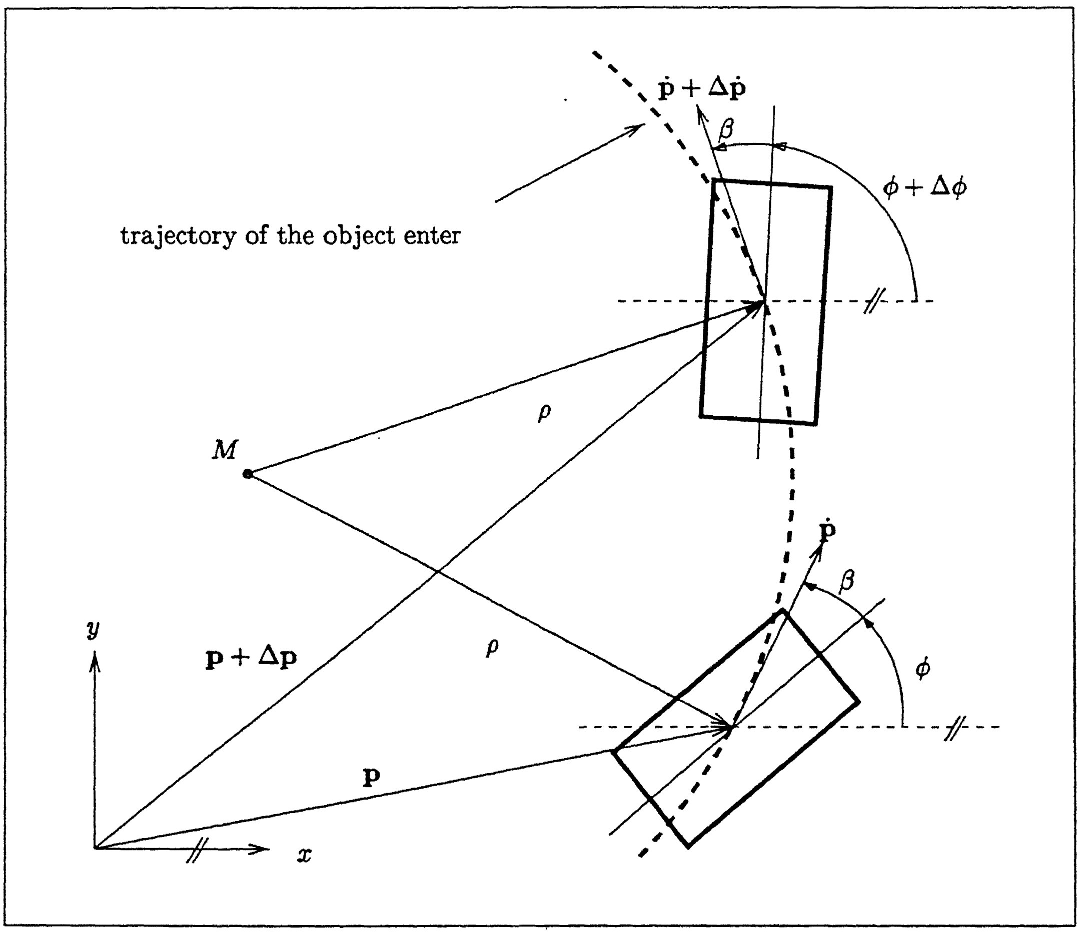


Figure 1.4: Stationary circular motion as a motion model. During the time interval $\tau = t_{k+1} - t_k$ the center of the object has shifted about Δp and rotated about $\Delta\phi$ about the normal (z-axis) (from [Koller 92]).

The position of the object center $\mathbf{p}$ in the street plane is described by Fig. 1.4:

$$\mathbf{p}(t) = C + \rho \begin{pmatrix} sin\,\phi(t) \\ -cos\,\phi(t) \\ 0 \end{pmatrix} . \tag{1.1}$$

Differentiation of (1) with respect to the time t and elimination of the radius ρ by using

$$v = |v| = |\dot{p}| = \rho\dot{\phi} = \rho\omega$$

results in the motion model described by the following differential equation

$$\dot{p}_x = vcos\,\phi\,; \quad \dot{p}_y = vsin\,\phi\,; \quad \dot{v} = 0\,; \quad \dot{\phi} = \omega\,; \quad \dot{\omega} = 0 \quad . \tag{1.2}$$

In equation (1.2) it is assumed that the principal axis of the model through the model center is tangential to the circle that is described by the moving object center. In general, this is not true, but the deviation – the so called slip angle β – could easily be compensated by shifting the center of rotation along the principal axis of the model to a position at which the principal axis of the model is tangential to the circle along which the vehicle drives.

From the above discussion, the primary input required for the motion model will be the speed with which the vehicle has to drive. This speed will be provided by decoding conceptual descriptions.

Chapter 2

Literature Survey

Many research groups are working in the area which links Computer Vision and Artificial Intelligence. In this review, we initially focus on current research work at Karlsruhe. The work of other researchers which are related to our goal, is also discussed briefly. The research work is classified into the following 3 categories.

- Optical Flow,
- Tracking,
- Natural Language Description.

Though the above 3 topics are interrelated, each topic is discussed separately in the following sections.

2.1 Optical Flow

Optical flow – the apparent shift velocity of grey value structures - is usually estimated by exploiting the postulate that the image intensity corresponding to a moving scene point remains temporally constant in a short image subsequence. [Horn 86] expressed the above postulate by an equation called *Optic Flow Constraint Equation* (OFCE): $g_x u_1 + g_y u_2 + g_t = 0$, where g_x, g_y, g_t are the partial derivatives of the gray value function and $(u_1, u_2)^T$ denotes the OF vector to be estimated.

[Otte & Nagel 95] studied local differential techniques in order to estimate optical flow and its derivatives based on the brightness change constraint. They analyzed two approaches, namely Neighborhood-Sampling (NS) and gray-value gradient Taylor Expansion (TE) for estimating the optical flow. In the former approach, they considered a spatial region $n \times n$ pixels around the image frame location $\boldsymbol{x}$ to sample the gray

value gradient at n^2 positions. In the latter approach, they described the gray-value gradient as a Taylor series.

[Kollnig & Nagel 96] incorporated the optical flow field into the measurement function for 3D pose estimation which uses the *inter-frame* information in addition to *intra-frame* information. They reported that using only image gradients, the position can be estimated accurately. Similarly, using only optical flow, orientation can be estimated accurately. But by combining image gradients and optical flow into the measurement function, position as well as orientation can be estimated satisfactorily. They concluded that with this approach, severely occluded as well as low contrast vehicle images could be tracked successfully.

[Nagel & Haag 98] noticed a lag which increases as the scene distance covered increases, between actual and estimated vehicle positions while incorporating **OF** into the measurement function. They analyzed and reported the reason as the underestimation of **OF** magnitudes. They introduced Bias-Corrected OF to minimize the lag. They concluded that by introducing bias corrected OF, the lag is significantly reduced.

2.2 Tracking

[Koller et al. 93] proposed an approach to detect and track road vehicles automatically. These authors used the following steps in their approach:

- Roughly identifying the moving region in the image by clustering moving image features which are projected back into the scene based on a calibration of the recording camera.
- Matching the straight-line edge segments extracted from the image with the 2-D model edge segments which are obtained by projecting a 3-D polyhedral model of the vehicle into the image plane. All matches are assessed on the basis of the Mahalanobis distance between data and model line segment attributes.
- An illumination model, which provides in combination with the vehicle model, a geometrical description of the shadows of the vehicle, is used to steer the matching step.
- The motion parameters for the motion model, which are essentially required for tracking, are estimated using a **MAP** estimator.

Establishing the correct correspondence between data and model segments is really a difficult task. To avoid the difficulty, [Kollnig & Nagel 97] developed a new

approach in which the gray value gradient of image features has been exploited. Here a synthetic image gradient, obtained by computing the gradient of the view sketch by means of its convolution with a bivariate Gaussian, is matched directly to the current image gray value gradient magnitudes. The difference is used to update the 3D pose of the model by MAP estimation. Finally a Kalman filter is used for tracking. They concluded that, with this approach, tracking can be done successfully even while the vehicle is partially occluded by foliage.

Combining the Gray Value Gradient and Optical Flow approach into the measurement function has been carried out by [Kollnig & Nagel 96] which is already discussed in the previous section.

Finally, [Haag & Nagel 98b] investigated an approach in which the advantages of taking both optical flow and edge element *orientation* into account have been combined. The authors applied one and the same approach with the same parameter set to track as many vehicles as possible in different image sequences rather than optimizing a tracking approach for a particular vehicle. In this manner, these authors succeeded in tracking 34 objects successfully while 17 objects were tracked acceptably. But they failed completely to track 5 objects due to the initialization problem as well as due to occlusion.

2.3 Natural Language description

Because of direct relation to our work, this section is discussed somewhat more elaborately. From 1977 onwards, different approaches had been studied for the extraction of conceptual descriptions from image sequences. [Nagel 88] suggested in his approach that it is essential to introduce a 'generically describable situation' for representing the complex discourse world. Moreover, the author insisted that the generic description for an agent should comprise the following 3 principal components:

- A generic description for its spatial structure, including position and orientation of an agent with respect to some reference frame.
- A generic description for all its motion states, i.e. for all components of the agent, the linear velocity and acceleration vectors of the center of gravity and the angular velocity around an axis through the center of gravity.
- A generic description of all 'intentions', i.e. goals of the agent.

As a consequence, [Nagel 91] introduced a series of levels which are,

change - deviation in a sensory signal which significantly differs from noise

event - any change which has been defined as a primitive for the construction of more complex descriptions

el_act_verb - characterizing an elementary activity such as 'to move'

load_act_verb - characterizing loaded activities such as 'to turn'

adverb_exp_verbphrase - qualifying attribute appears explicitly in the description such as 'driving fast'

verb_phrase - explicating adverbs as well as objects to which the verb relates

history - entire description which consists of *episodes* as the building blocks.

The above series of levels is used to extract an activity description from image sequences. Moreover, the author modelled goal-oriented behavior of an agent by a hierarchical set of *gd_situations* which describes the activities or actions of an agent. These actions in sequence are taken to represent behavior. Behavior are considered to follow plans which aim at achieving goals. Transition diagrams which consist of either non-terminal or terminal nodes, are used in order to represent a sequence of activities.

Extending the above approach, [Nagel et al. 95] used the concept of a hierarchical situation graph originally developed by [Krüger 91] which is the expansion of the transition diagrams mentioned above. A situation graph consists of 3 types of nodes namely *situation nodes*, which represent a situation, i.e. an association between action and state of the agent, *link nodes*, which represent a transition between a current and a successor situation, and *argument nodes* which are connected exclusively to a link node, linking the state of the current situation node to the corresponding state of the successor situation node. A situation node will be instantiated if there is a match between data for primitive concepts extracted from a halfframe image and the condition required for instantiation. The authors implemented such a hierarchical situation graph for a gas station scenario.

[Gerber & Nagel 96b] tried to generate more global conceptual descriptions by *quantifying* the occurrences using natural language quantifiers. The authors categorized the occurrences, which comprise the information about examined vehicles, the motion verb, and the validation time, into four classes which are *agent reference, road reference, object reference, and location reference.* Moreover, the authors used the Discourse Representation Theory (DRT) developed by [Kamp & Reyle 93] to generate quantified occurrences from the single agent occurrences by a set of derivation rules which are based on the following 3 postulates:

- A vehicle either moves or is standing.
- Each moving object is related to one occurrence of each class at each instant of time.
- At each instant, the state of an object is defined by at most one occurrence of each class.

[Haag et al. 97b] attempted to generate a conceptual description by using Fuzzy Metric Temporal Logic (FMTL) programming language - F-Limette - developed by [Schäfer 97]. This approach is able to represent and process uncertain, time-related data. FMTL uses tableau calculus and provides several different inference strategies such as depth search, breadth search, beam search compared to the logic programming language PROLOG which is based on resolution calculus and depth search. The authors reported the advantage of FMTL programming over PROLOG by implementing the beam search method for cases failed due to uncertainties in the estimated geometric data or due to several possible alternative behaviors which misleads the sequence of situation nodes, if only a single interpretation path is considered.

[Haag & Nagel 98a] extended the above approach with the new concept of *incremental* recognition of a traffic situation. They divided the system behavior into two layers namely the *geometrical layer* and the *inference layer*. In the former one, a geometric scene description, i.e. states of moving vehicles, is updated and corresponding fuzzy attributes are produced at each halfframe time point. Spatial relations will be computed using the estimated vehicle position, a lane model, and a 3D scene model. The results are provided to the inference layer, i.e. to F-Limette, which provides the mechanism of dynamic predicates in order to support the incremental evaluation. In this approach, data provided by the geometrical layer is processed immediately by the inference layer. In the previous approach, geometrical and inference tasks were separated, i.e. generating first a complete geometrical scene description for an entire video sequence, followed by inference steps on its global set of data. Advantages of using this incremental recognition have been reported.

In contrast to the above approaches [Howrath & Buxton 98] proposed two ways to generate conceptual description. The first, called "monitoring", uses little top-level control instead of following the flow of data from images to interpretation. The second, called "watching", emphasizes the use of top level control and actively selects evidence for task based description of the dynamic scenes.

Most of the approaches discussed above give the output in conceptual description form. An attempt has been made by [Gerber & Nagel 98] to derive the natural language text from the conceptual description. They devised three layers for their work

namely Signal Layer (*SL*), Conceptual Layer (*CL*), and Natural Language Layer (*NL*). SL provides the geometric results by a model-based image evaluation system, *fuzzy metric temporal* predicates, and conceptual primitives such as motion, road, occlusion and spatial primitives which are all required by the CL. CL generates the 'facts' by traversing a Situation Tree. NL transforms the results obtained from CL into 'Discourse Representation Structures (DRS)' which facilitates the derivation of natural language text by Computational Linguistics based on a number of heuristics.

[Mukerjee et al. 98] attempted to generate the visual scene and actions within this scene from the input which is based on conceptual descriptions. To accomplish that, they concretize the conceptual model which can capture different variabilities inherent within the scene by creating a large visual database of objects and actions, along with a set of constraints which are combined with multi-dimensional fuzzy functions called continuum fields. An instance is generated by identifying minima in the continuum fields which are used to create default instantiations of the objects described. The resulting image may be considered to be the "most likely" visualization, and if this matches the linguistic description, the continuum fields selected are a good model for the conceptual content in the linguistic model of the scene.

[Gupta et al. 98] analyzed the behavior of a mobile robot which may be controlled by a set of schemes based on potential behaviors. Dynamic effects were controlled by constraining the range of possible motions as a function of velocity. Moreover, each agent has a set of beliefs regarding its own behavior such as aggressiveness, speediness etc., and the behavior of other vehicles which is inferred by modeling through a projective potential model where the vehicles are projected to their likely future positions. Finally, these authors concluded that with this approach conventional path–planning schemes involving explicit search can be replaced by local decisions using a potential field. But at the same time, the authors mentioned that a potential field can not be used to execute pre-planned motion sequences such as Parking or "U"-turning in narrow roads.

Chapter 3

Primary Tasks

In order to progressively achieve our goal mentioned in the abstract, we have divided the goal into the following 3 subtasks:

- Projecting a polyhedral model onto an image - for testing purposes.
- Maneuvering the specific vehicle on the specified lane. This requires the following two subtasks:
  - Decoding the conceptual description. The output will be a vehicle number, the lane number on which the vehicle is moving, and its speed. These will be the input for the motion model which is used for maneuvering the vehicle.
  - Loading the lane structure representation into the system which requires to accomplish the previous task.
- Comparing the synthetic image sequences(**SIS**) obtained by the above mentioned tasks with original image sequences(**OIS**) taken by a real static camera.

The above 3 subtasks have to be carried out for each of the different cases mentioned in the introductory chapter. Here, we are reporting the first subtask of projecting the polyhedral vehicle model on to the static gray value image at the specified location, based on a virtual camera position. The way by which we have accomplished the above subtask is explained briefly in the following sections.

3.1 Loading the Image

The picture manipulation (pm) file, which represents the gray value image of the static scene, is loaded onto a window-like screen interactively. This loaded image

will be considered as the background image on which experiments have to be carried out, i.e. projecting the lane structure, vehicle model and maneuvering the vehicle on the specified lane. Such an image representing an intersection scene is shown in Figure 3.1.

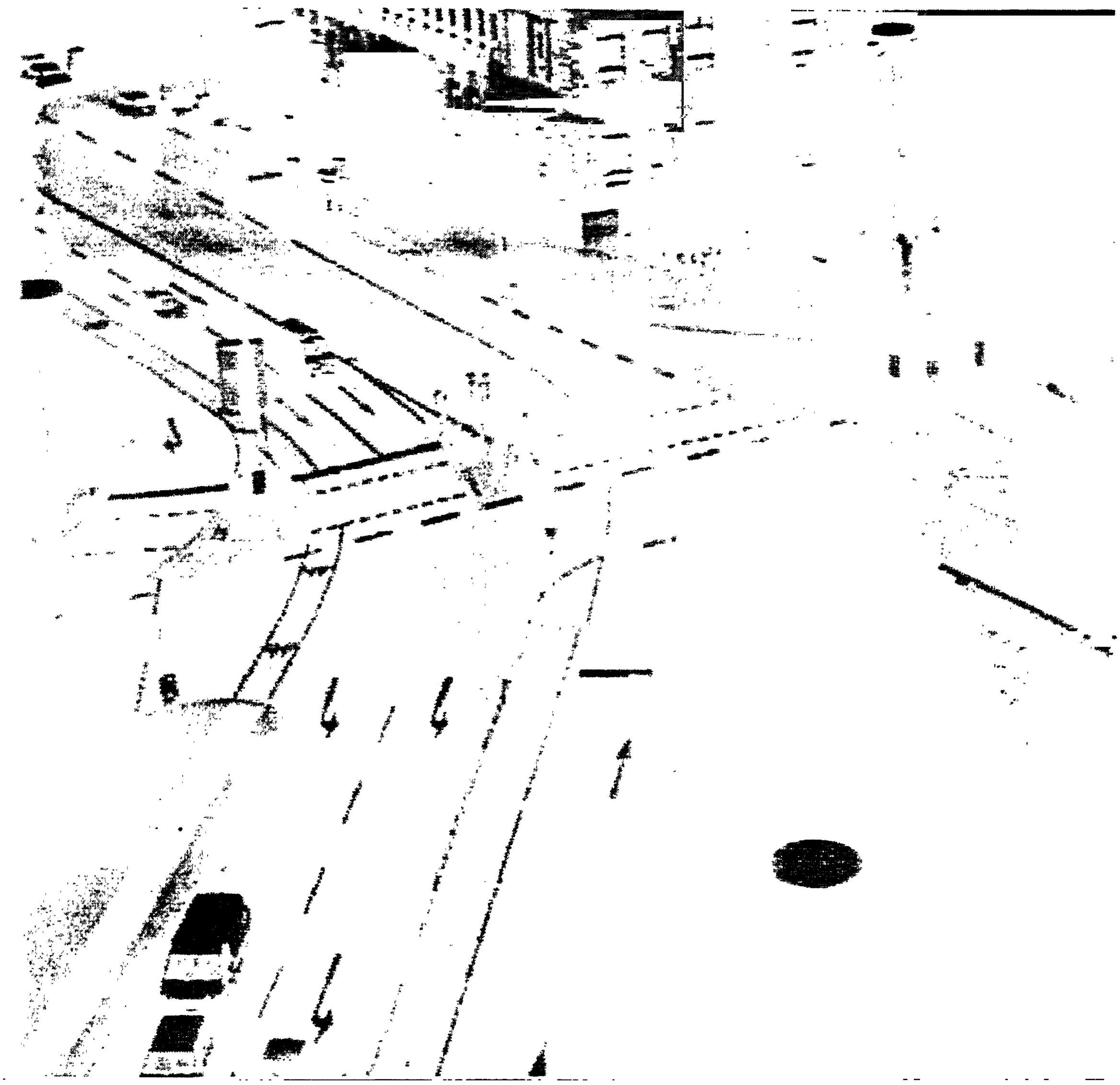


Figure 3.1: Image of the kwbB –sequence, which represents the intersection scene and will be considered as the background image for our experiments

3.2 Vehicle Model

The file, which describes the specific vehicle model, contains 12 length parameters (Table 1.1) and the generic description of edge points which are based on a model coordinate system. We have taken the model coordinate system as the center point of the vehicle. Different instantiations of the vehicle model can be obtained by varying the values of the 12 length parameters. For testing purposes, the vehicle model is chosen interactively. But for the actual work, this should be obtained by exploiting the input from conceptual description.

3.3 Model to Scene Transformation

The scene representation corresponds to a top view of the road section. Because the vehicle will be maneuvered on the road surface, each edge point of the vehicle model should be represented in the scene-based coordinate system. This is done by using the usual matrix multiplication which is given by

$$\mathbf{P}_s = R_s^m \mathbf{P}_m + \mathbf{t}_s^m$$

where $\boldsymbol{P_s}$ represents the vehicle edge points in the scene coordinate system and $\boldsymbol{P_m}$ represents the model coordinate system. R_s^m represents the 3×3 rotation matrix and $\mathbf{t}_s^m$ represents the translation parameters

Out of 6 parameters, only 3 parameters which are the translations along the x and y directions and the rotation about the z axis, are required for inferring the vehicle pose. These parameters should be obtained from conceptual descriptions. For testing purposes, we have provided these parameters interactively.

3.4 Scene to Camera Transformation

This conversion is required to project the vehicle model onto the image plane. We have implemented the concept of a *virtual camera*, i.e. moving the camera around the vehicle. This facilitates us to get the different views of the vehicle in the static background scene. This is done, interactively, by changing the camera parameters which includes 3 rotation and 3 translation parameters as well as its focal length. With a specific parameter set obtained by calibrating the real camera, the vehicle model can be projected on to the corresponding gray value image which can be used

for comparison, i.e. whether the vehicle is projected at the expected position and orientation or not. The following formula is used for the above mentioned conversion.

$$\mathbf{P}_c = R_c^s \mathbf{P}_s + \mathbf{t}_c^s$$

where $\boldsymbol{P_c} = (x_c, y_c, z_c)^T$ represents the vehicle edge points with respect to the camera coordinate system. R_c^s represents the 3×3 rotation matrix and $\mathbf{t}_c^s$ represents the camera translation parameters.

3.5 Camera to Image Transformation

Perspective projection is used in order to project the 3D vehicle model on to the 2D image plane. This is done by the following simple formula:

$$x_i = f * x_c / z_c,$$

$$y_i = f * y_c / z_c.$$

where (x_i, y_i) represent the position of vehicle edge points in the image plane and f indicates the focal length of the camera.

3.6 Next task

Loading the lane structure allows to superimpose the background image by the projection of the lane structure on to the image plane. Because the output of decoding conceptual description will be a vehicle and the lane rather than the exact position and orientation of the vehicle. Once we have the loaded lane structure, the vehicle will be projected on to the specified lane. Moreover, from the geometric description of the lane structure, the vehicle will be positioned at the center line of the lane, with the orientation parallel to the longitudinal axis of the specified lane.

Chapter 4

Straight Lane

We have achieved the first milestone in our goal, i.e. the first simple case of maneuvering a single vehicle on a single and straight lane. The following sections briefly discusses the procedure by which we have accomplished the task.

4.1 Loading Lane Model

The traffic scenes such as straight road sections or intersections etc. consist of several lanes which may be straight or curved. The entire lane structure projected onto the image plane and superimposed onto the image is shown in Figure 4.1. Exploiting the entire lane model may be required to accomplish the complex task. So to carry out the experiment for the simple case, we have considered only two lanes which are essentially straight.

A file which consists of the edge points of each segment of each lane, is converted into an internal representation from which the lane structure is projected onto the image plane. Moreover, we have divided each lane into several segments such that each segment will be as straight as possible. This will implicitly facilitate to maneuver the vehicle on a curved lane also. Loading the lane model is essential in order to maneuver the vehicle on a specified lane. More specifically, we can say, to get an initial pose of the vehicle.

4.2 Conceptual Description

As we have mentioned in the abstract, conceptual descriptions are used in order to generate the SIS of vehicle maneuvers. Such a description contains various terms

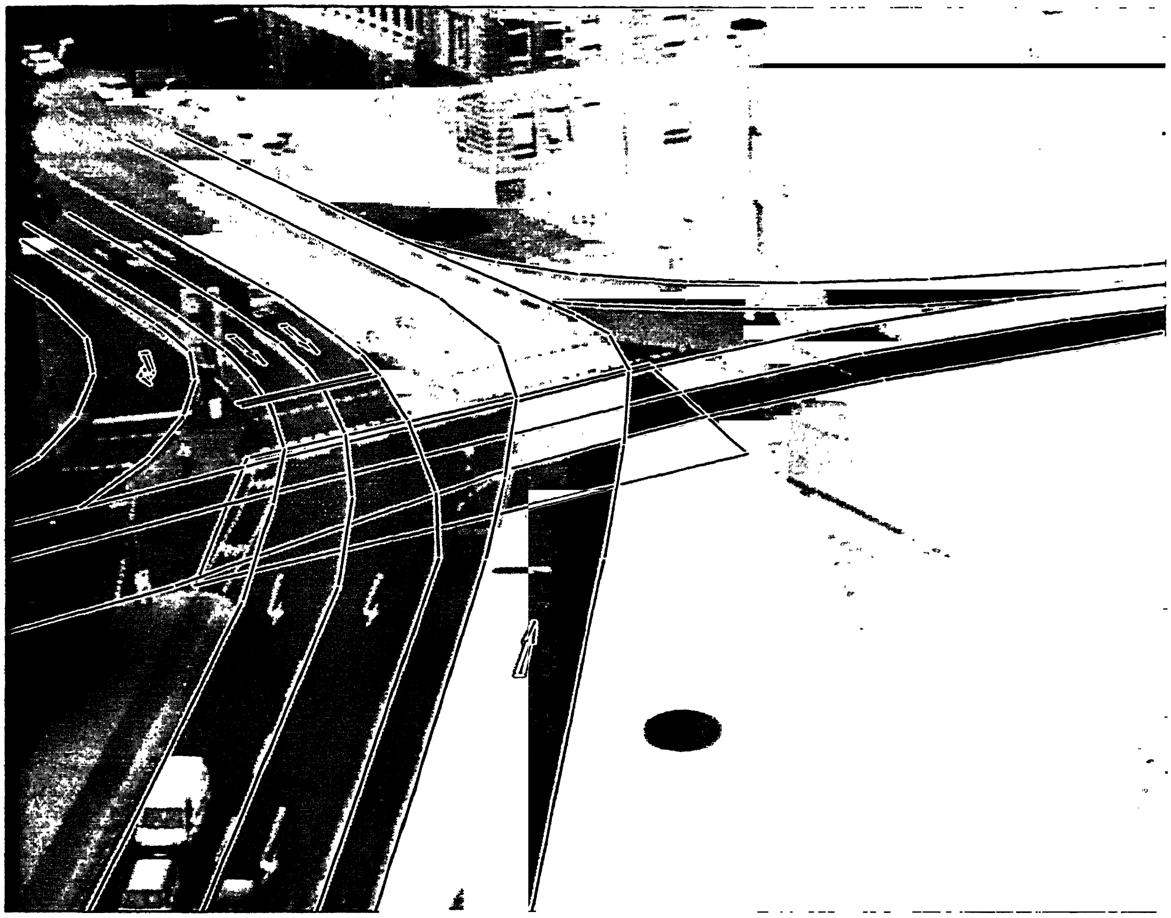


Figure 4.1: Complete lane model projected onto the image plane and superimposed onto the background image of Figure 3.1.

which are related to a car maneuver. A file representing the conceptual description consists of a set of FMTL predicates and number of rows which depend on the time limit within which the vehicle is tracked by the Xtrack system.

```
0.5 | 2277 : 2278 ! speed(obj_1, veryslow)
0.7 | 2380 : 2381 ! speed(obj_1, normal)
0.5 | 2380 : 2381 ! speed(obj_1, slow)
0.3 | 2480 : 2481 ! mode(obj_1, forward)
0.4 | 2693 : 2694 ! mode(obj_1, stand)
0.8 | 2110 : 2111 ! on (obj_1, lane_1)
```

The above few lines show how the conceptual description will look like. The columns represent respectively the degree of validity, the time interval, and the predicate description which may be 'speed', 'mode' or 'on'. The first argument of each predicate represents the vehicle which has been tracked. The second argument may represent a speed characterization such as 'normal', 'slow', 'very slow', and 'null' etc. or for a predicate that refers to mode of driving. It may represent either 'forward', 'backward' or 'stand'. If the predicate is 'on', its argument will be the lane on which the vehicle is driving.

All predicates except the predicate 'mode' has been exploited in the straight lane model.

4.3 Getting an Initial Vehicle Pose

The procedure for loading the vehicle model is already discussed in the previous chapter. Previously, the file representing the vehicle model and the principal components, i.e. the pose at which the vehicle will be projected, have been given interactively. Both inputs are now obtained from a given conceptual description.

First, the program looks for the predicate 'on' in the conceptual description to get the lane label number. This will indicate merely the lane on which the vehicle will be driving rather than a detailed initial pose. The assumption is that the vehicle always starts from the first segment of the specified lane. Moreover, we have assumed that the starting position will lie on the middle axis of the first segment of the specified lane and the vehicle orientation will be parallel to the longitudinal axis of the lane and that its speed will be as per the conceptual description.

4.4 Maneuvering the Vehicle

For maneuvering the vehicle, we have used a circular motion model which was discussed briefly in the introductory chapter. After getting the initial pose, successive poses are calculated to maneuver the vehicle by the following formula:

$$\begin{aligned} x(t+1) &= x(t) + v(t) \cdot cos\, \phi(t) \cdot \Delta t \\ y(t+1) &= y(t) + v(t) \cdot sin\, \phi(t) \cdot \Delta t \\ \phi(t+1) &= \phi(t) + \omega(t) \cdot \Delta t, \end{aligned}$$

where, $(x(t+1),\ y(t+1))$ refer to the new position of the vehicle, $(x(t),\ y(t))$ refer to its old position , $\phi(t+1)$ and $\phi(t)$ refer to the new and old orientation of the vehicle, respectively. $v(t)$ represents the speed and $\omega(t)$ represents the angular velocity of the vehicle.

At each half frame time point, there is a possibility of different predicates related to the vehicle speed due to the fuzzy modeling of the speed characterization. We have chosen the predicate which has the highest degree of validity to simplify this problem. Finally, the value for the speed is provided by decoding those terms which represent the speed in the conceptual description. Constant magnitude has been taken for each velocity term. Because we have restricted our experiment to maneuver the vehicle on a single and straight lane, the orientation of the vehicle is also constant in magnitude.

4.5 Results

We have validated our developed system by testing it with 4 different objects. All 4 objects will be maneuvered on two different lanes which are essentially straight. The results, which are produced by our system, are given below.

Figure 4.2 shows the appropriate initial pose of objects 1 and 2. Notice a high magnitude lag between the object which we projected by our system and the corresponding object in original image sequences taken by a real static camera. This kind of lag is noticed for the other two objects also, when those two are projected. The reason for this lag may be due to unavailability of getting an exact initial pose for the object from conceptual descriptions. This lag could be eliminated by giving exact initial coordinates which are not 'conceptual' and are not available. The assumption has been made, therefore, that the vehicle will always start from the initial segment of the specified lane.

Figure 4.3 shows two poses of object 1 at various time frame points obtained from our **SIS**. A significant lag always exists between the object position in the synthetic and in the original image sequence.

Figure 4.4 shows object 4 at two different poses which correspond to the pose at half frame time point 1200 and at half frame time point 2200, though the corresponding object in the original image sequence stands and thus remains in the same position. This is being noticed for other objects also. We have not yet exploited 'stop' and 'start', which are present in the **OIS**, in the form of conceptual descriptions. The term representing the minimum speed characterization has been taken even when those predicates arise. This causes the object to maneuver always forward without stopping.

4.6 Discussion

Our developed system satisfies an essential requirement for maneuvering the vehicle, i.e. to keep its pose always in a lane on which it is moving. Note however that the **SIS** produced by our system do not match with the original image sequences. Partly this is because in the straight lane model, we ignore some attributes such as 'stop' in 'mode' predicate, which is otherwise useful. However, to provide a better match the conceptual description will have to be enhanced, for example by providing the exact initial pose of the vehicle.

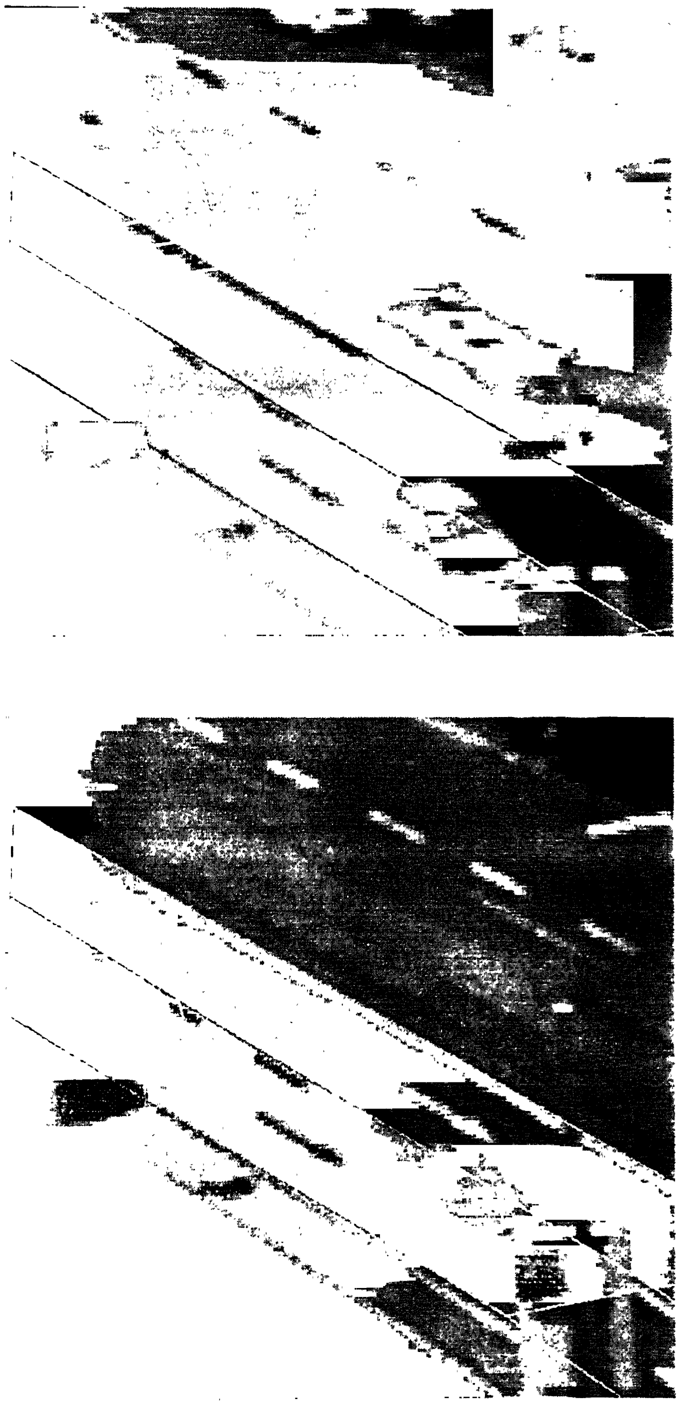


Figure 4.2: The initial pose of object 1 (top) and 2 (bottom) at half frame time point 2120 and 40 of the kwbB –sequence, respectively.

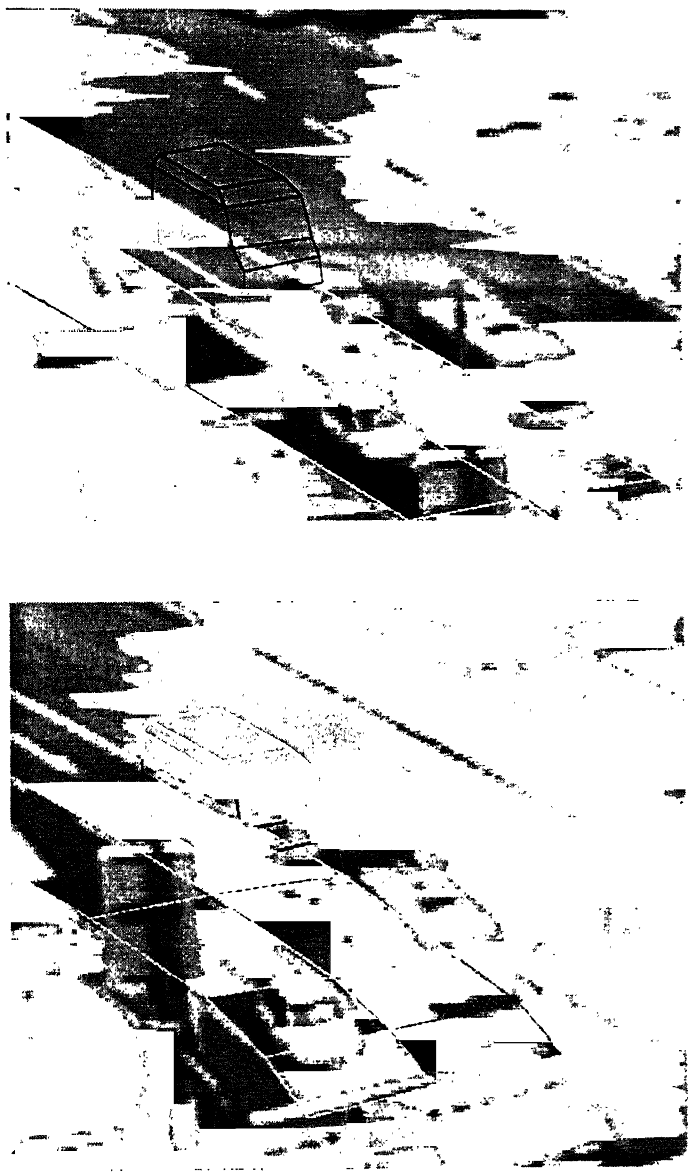


Figure 4.3: The pose of object 1 at half frame time point 2400 (top) and 2650 (bottom). Both panels indicate that a significant lag always exists between **SIS** and **OIS**.

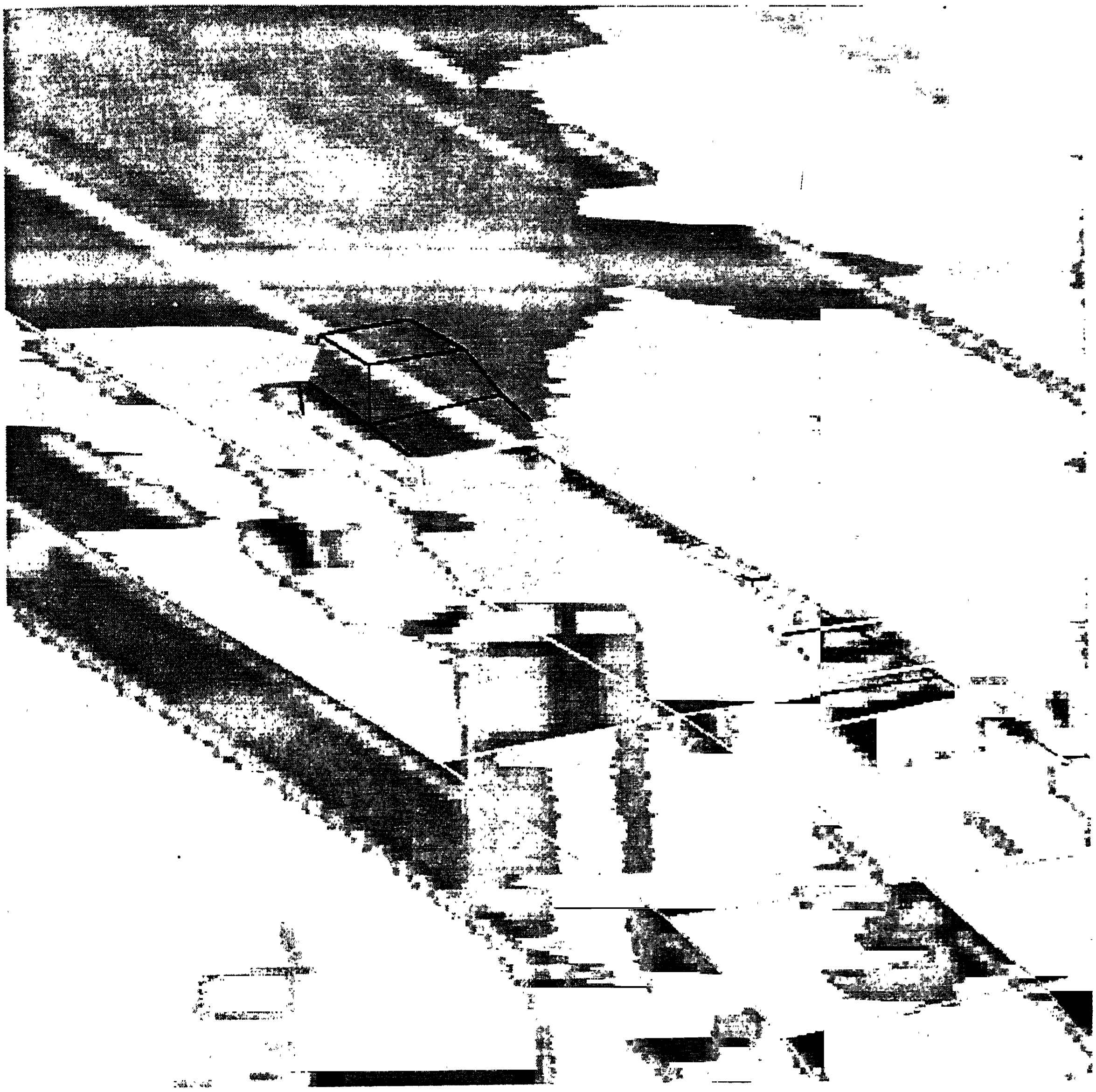


Figure 4.4: The pose of object 4 at half frame time point 1200 and 2200. This shows that the object in **SIS** moves forward though the corresponding object in **OIS** stops and remains standing in that position.

Chapter 5

Turning

The next task maneuvering the vehicle on a curved lane is reported here. In this task, the primary aim is to always keep the vehicle within the lane on which it has to be maneuvered based on the input obtained from conceptual descriptions. In order to accomplish this, three motion models have been tried which are

- Steering Angle Motion model,
- Circular Motion model with constant angular velocity,
- Circular Motion model with varying angular velocity.

Each of the above models is discussed briefly in the following sections. In addition to this, we have exploited the 'stop' attribute in 'mode' predicate of the conceptual description and the vehicle which is going to be maneuvered will start from the center of the first segment of the initial lane instead of from the point which we used for previous cases. By including these two factors in our experiments, the lag between vehicles in synthetic and original image sequences is significantly reduced.

5.1 Steering Angle Motion Model

This model is more physically motivated. It considers the steering angle which changes during a turning maneuver. Using this model successful results on *tracking* have been reported by [Nagel et al. 98]. The Figure 5.1 shows the steering angle model in which successive poses of the vehicle are calculated as a function of the steering angle ψ as given below:

$$x(t+1) = x(t) + v(t) \cdot \cos(\phi(t) + \psi(t)) \cdot \Delta t$$

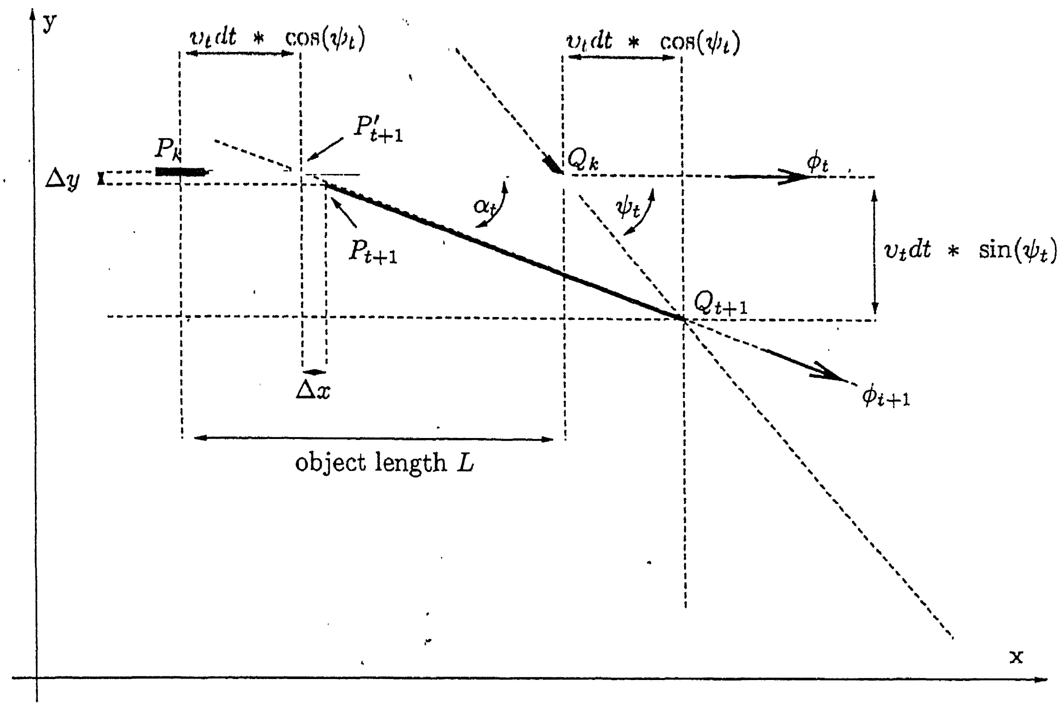


Figure 5.1: Configuration of a vehicle moving on a circular path at time point t and $t+1$. P and Q denote the reference points of the front and the rear wheel, respectively, ψ denotes the steering angle, and ϕ the orientation of the vehicle (from [Nagel et al. 98]).

$$\begin{aligned} y(t+1) &= y(t) + v(t) \cdot \sin(\phi(t) + \psi(t)) \cdot \Delta t \\ \phi(t+1) &= \phi(t) + \alpha \cdot \Delta t, \end{aligned}$$

where

$$\tan(\alpha) = \frac{\sin(\psi) \cdot v\,\Delta t}{\text{vehicle length}}.$$

Now the query arises at what position the steering angle has to be tuned and at what position it has to be reset to zero. To simplify this problem, an assumption has been made that the steering angle will be in effect only when the car is in between the positions that are three fourth of the current segment length and one fourth of the next segment length. Moreover, to make the experiment simple, we have assumed that the steering angle will be constant while turning.

5.2 Circular Motion Model with constant ω

We have considered a model which is simple compared to the previous model and which has already been discussed in the introductory chapter. Here, we made an assumption that the angular velocity – the temporal derivative of the vehicle orientation – of the vehicle is constant while it is maneuvering from three fourth of the longitudinal extension of the current segment to one fourth of the next segment. The following formula shows how the angular velocity is calculated:

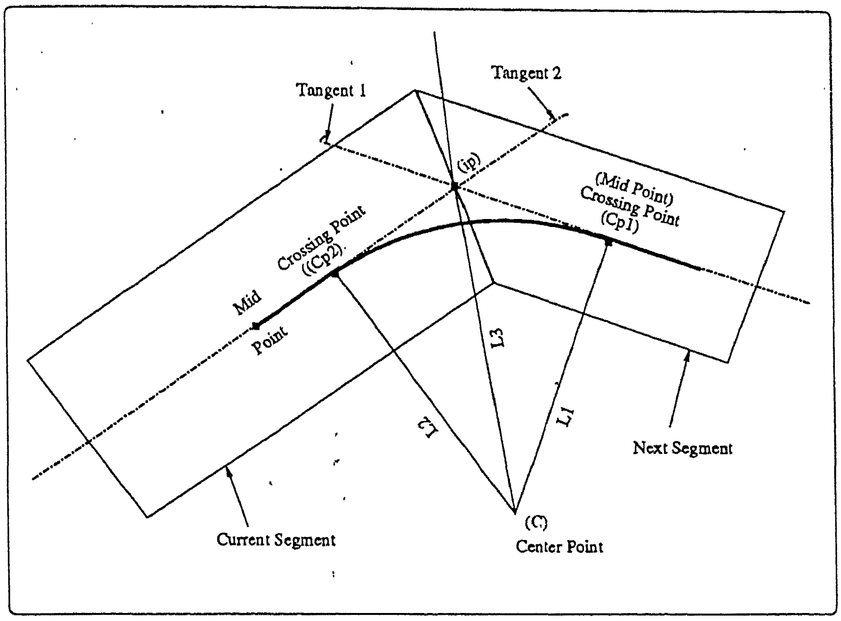
$$\begin{aligned} \omega &= \frac{\Delta\phi}{t}, \\ t &= \frac{d}{v}. \end{aligned}$$

where $\Delta\phi$ represents the difference between the target which is nothing but the orientation of next segment and the current orientation, t refers the approximate time required to attain the target orientation, d is the Euclidean distance between three fourth of the current segment and one fourth of the next segment, v is the speed at which the angular velocity is going to be tuned.

5.3 Circular Motion Model with varying ω

To overcome the drawback of the two previous models, we have developed a new motion model in which the angular velocity (ω) will be varied while maneuvering on a curved section. Instead of between three fourth of the current segment and one fourth of next segment as we assumed in the previous two models, the angular velocity will be in effect between the *crossing* points of the current and the next segment. Figure 5.2 shows how the circular motion model has been constructed. The following steps describes how the crossing points for both the segments have been calculated.

1. The crossing point for either one of the segments will be calculated based on the length of both current and next segment, i.e. if the length of the current segment is less than that of the next segment, the crossing point for the current segment will be its mid point (**Cp1**). Otherwise the mid point of the next segment will be taken as its crossing point.
2. Next, the center point (**C**) of the circular arc will be calculated by intersecting the two lines, namely **L1** which is the line perpendicular to tangent1 and passing through **Cp1**, and **L3** which is the line passing through the intersecting point



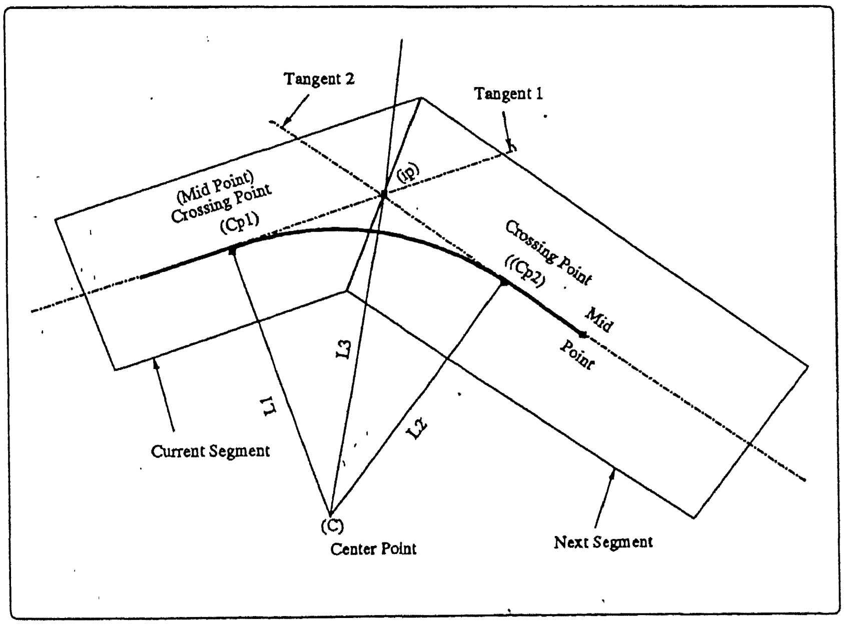


Figure 5.2:
Construction of circular motion model for the case of shorter (top) or longer (bottom) target segment. The base crossing point **Cp1** is located on the shorter segment at the midpoint. The angular bisector of the tow longitudinal axes, **L3**, is intersected with **L1** to obtain the center C. Consequently the other crossing point **Cp2** is equally spaced from the intersection point (**ip**).

(ip) of tangent1 and tangent2 and parallel to the middle orientation between the current and the next segment.

3. Finally, the other crossing point **(Cp2)** will be calculated by intersecting two lines namely tangent1 and **L2** which is the line perpendicular to the longitudinal of the next (or current) segment and passing through the center point **(C)**.

The formula which is used to determine ω at each half frame time point while turning is given below:

$$\omega = \frac{v}{r}.$$

where r represents the radius of the circular arc. By this method, we can assure theoretically that the vehicle will attain the target orientation along with the target position which ensures that the start and the target position of the maneuver will always lie on the longitudinal axis of the start and target lane, respectively.

5.4 Results

Firstly, the results obtained by each of the above discussed models are provided and reasons for the failure discussed. Next, the comparison between **SIS** and **OIS** is reported.

First we implemented the steering angle motion model for car maneuvers. The results obtained by this model for four different vehicles are shown in Figure 5.3. Notice, initially all vehicles keep their position with in the lane. But as soon as the vehicles start to turn, their position in **SIS** deviates from the middle axis of its lane. Though the vehicles attain their target orientation at each half frame time point while turning, the position of the vehicle is offset from the center axis of the lane. This initial error is propagated further which leads the object to deviate entirely from the lane at later stages. This motion model could be improved by moving the cross points closer to each other and varying the steering angle during a turning maneuver.

Figure 5.4 shows the trajectory of four vehicles obtained by using the circular motion model with constant angular velocity. By this model, we were able to keep object 1 and 3 within the lane while they are maneuvering. Object 2, however, deviates from the lane after it crossed nearly half the length of the whole lane and object 4 deviates when it starts to turn. The reason could be that the position of the vehicle falls behind or surpasses the expected position based on the speed obtained from conceptual descriptions while it is turning. If the speed is less at a successive time frame point while turning as compared to the speed at which the angular velocity is going to be

Finally we implemented a newly developed model in which the concept of varying angular velocity is used along with a circular motion model. Figures 5.5 and 5.6 show the trajectory and pose of four different objects at successive 50 half frame time points. With this new motion model, we were able to keep all vehicles in their lane during the whole maneuvering time, as compared to the previous two models.

Notice from Figures 5.5 and 5.6, all four objects initially decelerate, stop nearly at the stopping line and start to accelerate. This indicates that our system is able to generate synthetic image sequences comparable with the original image sequences.

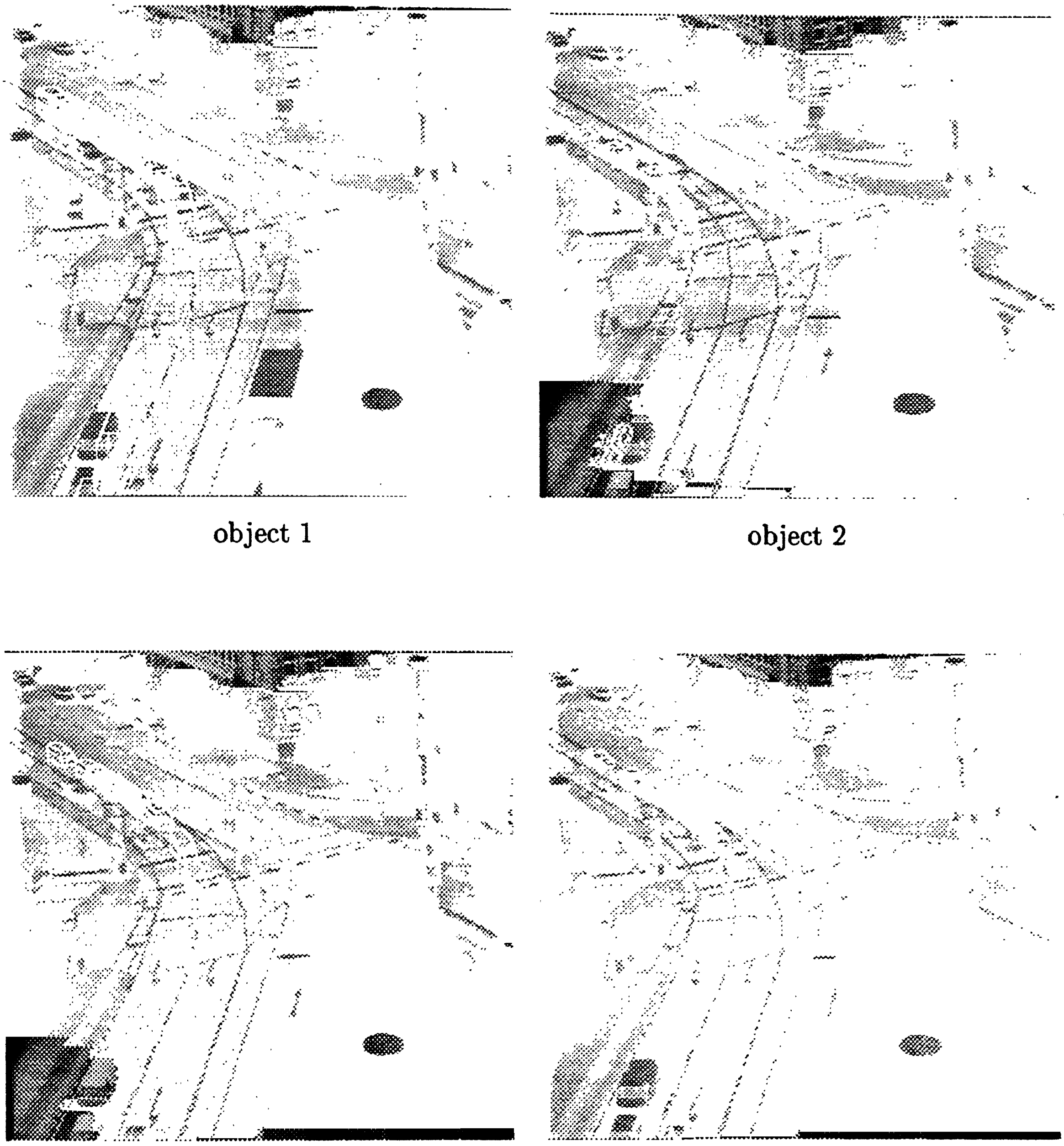
The top panel in Figure 5.7 shows the start pose of the object 1. As we mentioned earlier, all vehicles will start their maneuvers from the mid point of the first segment of the initial lane (see Fig. 5.2). This helps us to reduce the initial lag between the vehicle position in **SIS** and **OIS** which we noticed in previous cases. This is true for other objects also except for object 3 where the initial lag between **SIS** and **OIS** is still significant as shown in the bottom panel of Figure 5.7.

The top panel in Figure 5.8 shows the position of the vehicle at half frame time between 2330 and 2440. It means that the object 1 stops in front of the stopping line as in **OIS**. This is achieved by exploiting the 'stop' attribute in the 'mode' predicate in conceptual descriptions which allows us to generate synthetic image sequences in a more realistic way. This is being noticed for objects 3 and 4 also. But object 2 stops after it crosses the stopping line as shown in the bottom panel of Figure 5.8. This is due to the fact that the conceptual description does not report spatial landmarks such as stopping line.

The Figure 5.9 shows the pose of object 1 at 2700 and 2900 half frame time points, respectively. The pose of object 1 in **SIS** always surpasses the pose in **OIS**. The difference noticed between **SIS** and **OIS** increases at later time frame points. This is being noticed while doing experiments with other objects also. The reason could be that the value which we have assumed for the speed, is always constant and may be higher than the value which is used to obtain speed predicates. For example, the 'normal' attribute of the 'speed' predicate assume a value of 35 km/h while the actual value, which is used to obtain that speed characterizations by FMTL, varies from 10 to 60 km/h.

5.5 Discussion

It can be easily seen from the results obtained from three motion models that the circular motion model with varying angular velocity is giving better results compared to the other motion models. So, this model is chosen to continue our experiments.

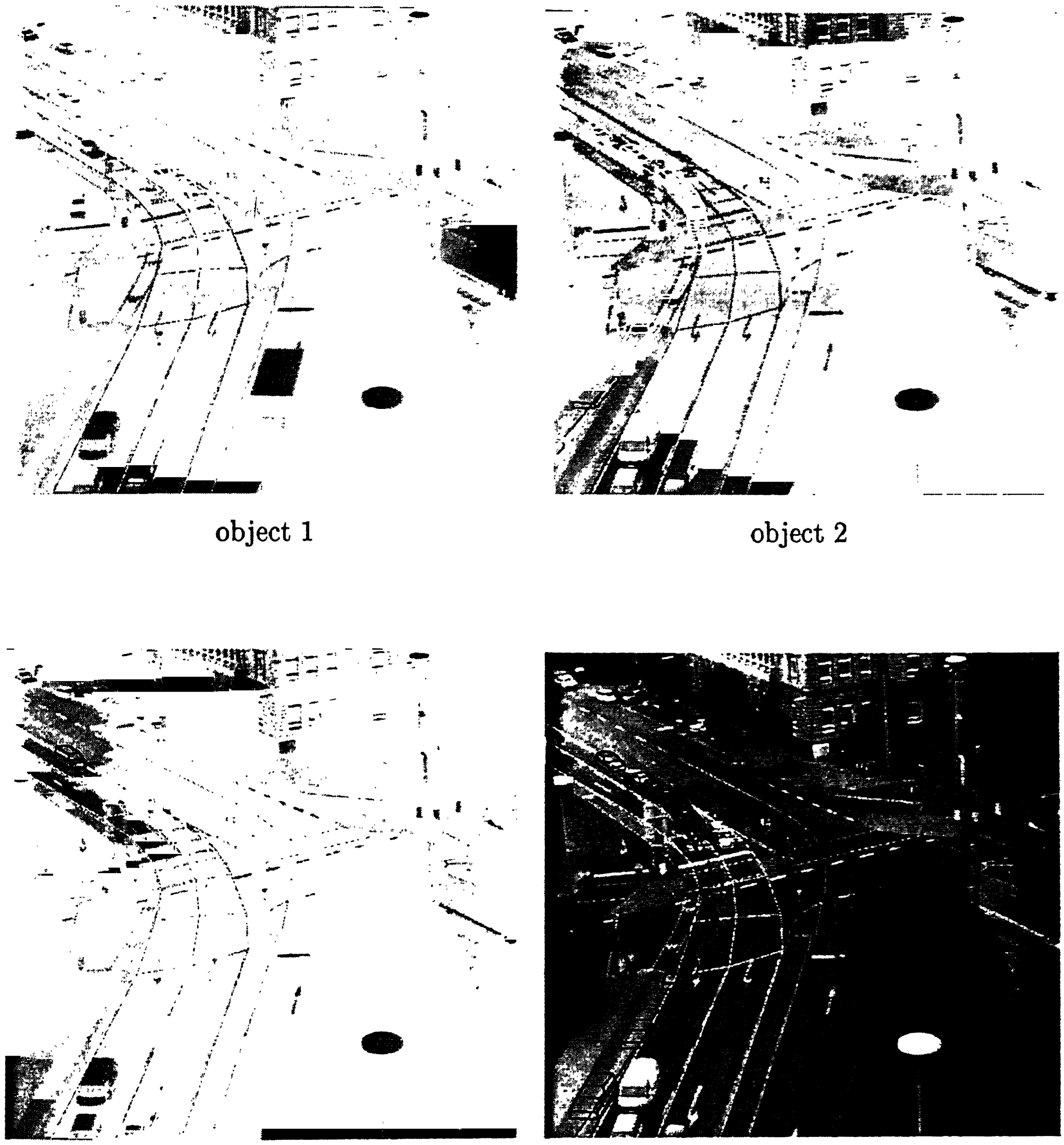


object 1 object 2

object 3 object 4

Figure 5.3: The trajectories of object 1, 2, 3 and 4 obtained by implementing a steering angle motion model.

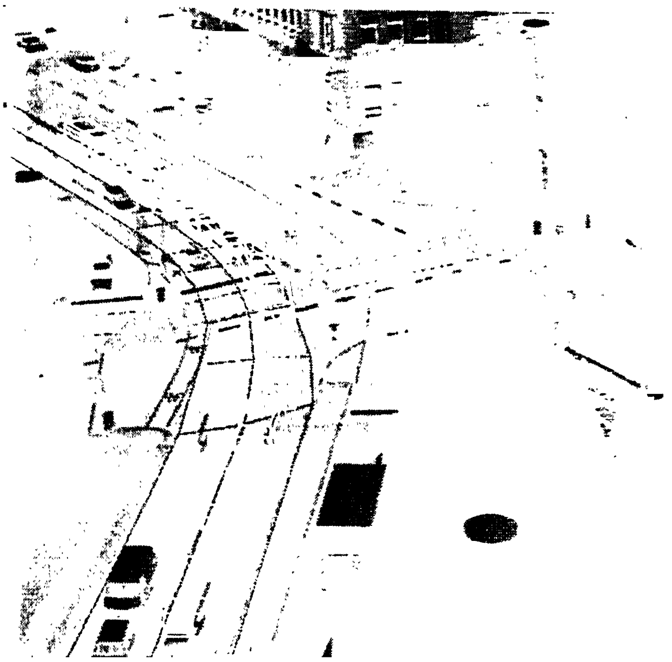
The reason which we cited in the previous section for the difference in position between **SIS** and **OIS** could be reduced by better velocity modelling eg. by using more than one fuzzy predicate and thereby smoothing the velocity over temporal intervals. This would also be easier if the fuzzy function is invertible - eg. it it is triangular



object 1 object 2

Object 3 object 4

Figure 5.4: The trajectories of object 1, 2, 3 and 4 obtained by implementing a circular motion model with constant angular velocity.



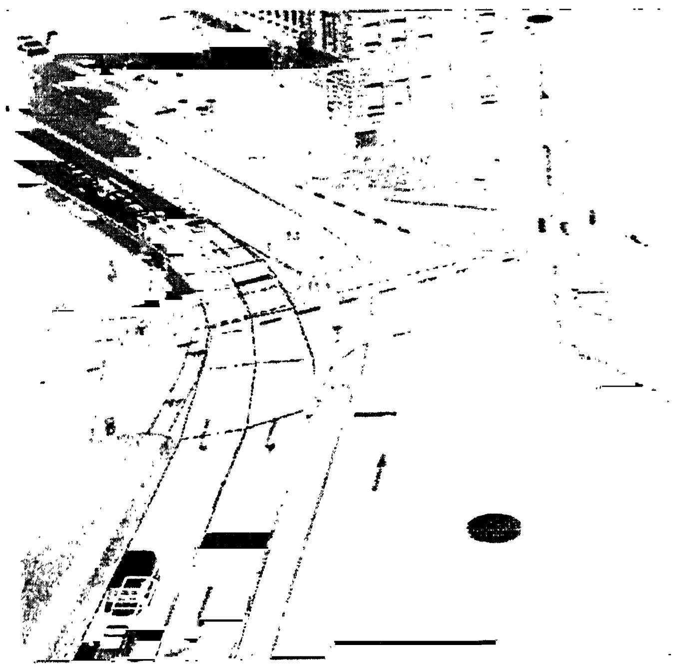


Figure 5.5: The trajectory of object 1 and 2 obtained by implementing a circular motion model with varying angular velocity.

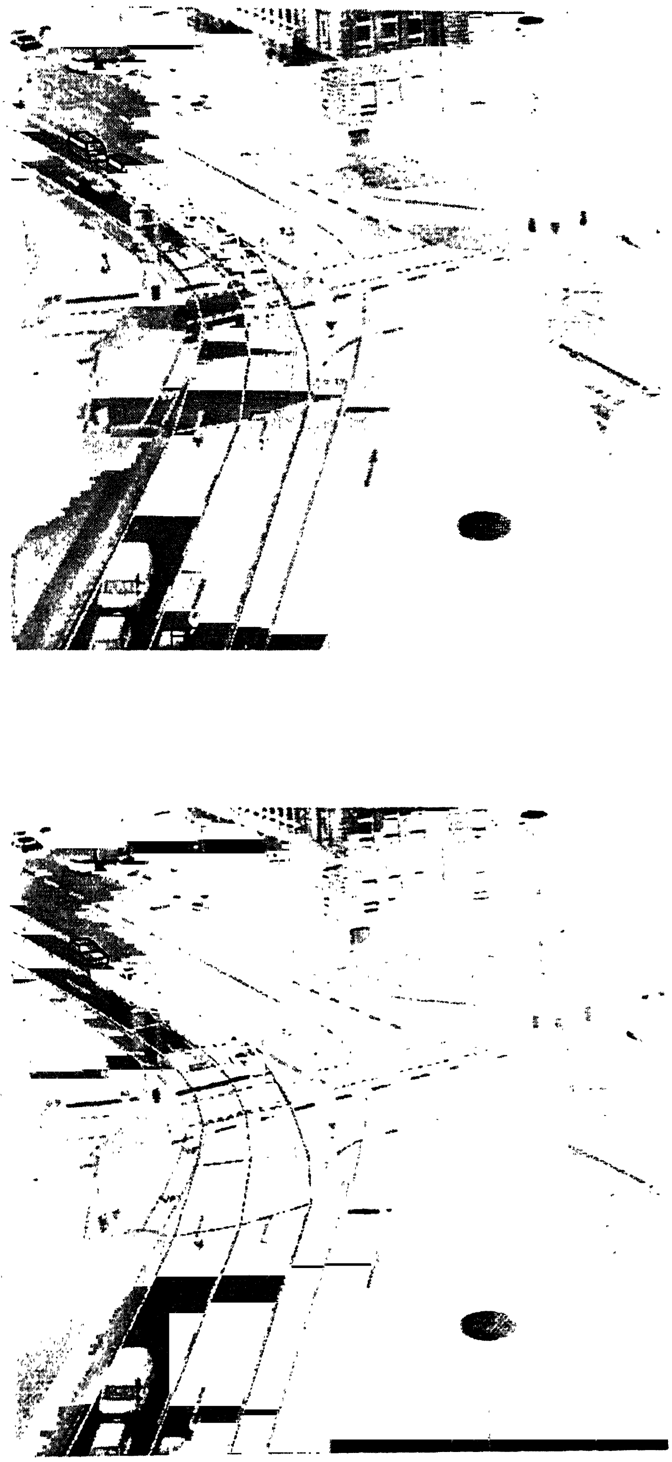


Figure 5.6: The trajectory of object 3 and 4 obtained by implementing a circular motion model with varying angular velocity.

Figure 5.7: The initial pose of object 1 (top) indicates that the initial lag is reduced when the vehicle starts from the mid point of the first segment of the initial lane. The bottom panel shows that the initial lag is still significant for object 3.

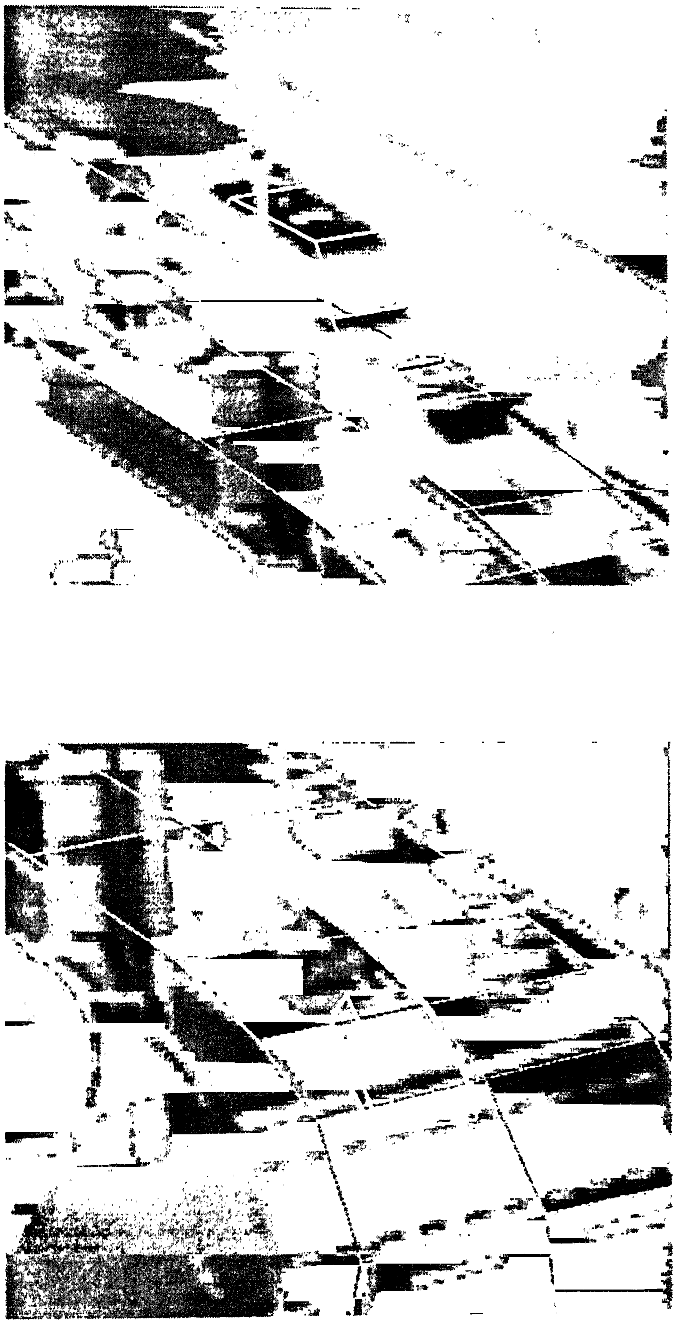


Figure 5.8: The pose of the object 1 at half frame time point between 2330 and 2440 (top) and the pose of the object 2 at half frame point between 300 and 2300. Both panels indicate that the object stops nearly at the stopping line.

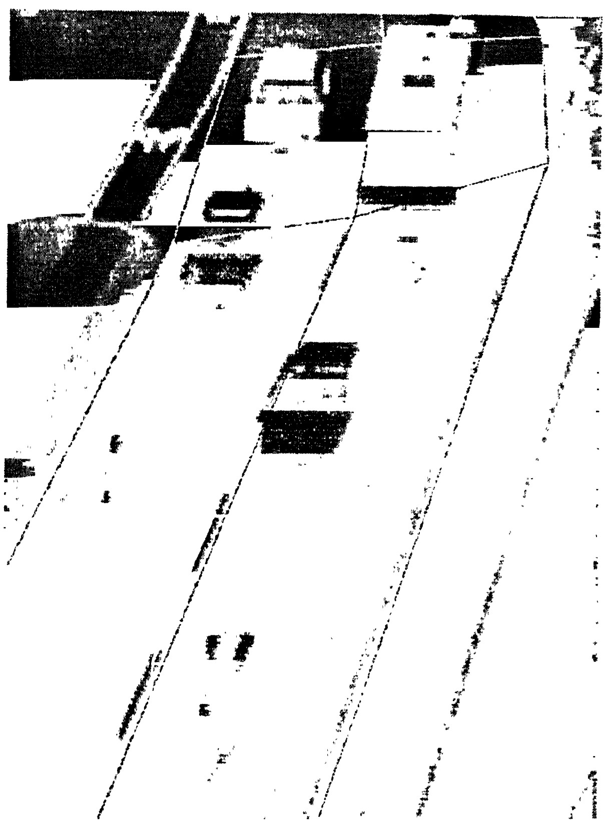


Figure 5.9: The pose of object 1 at half frame time point 2700 (top) and 2900 (bottom) show that that at later time frame points, the noticed difference between **SIS** and **OIS** increases

Chapter 6

Car Following

In this chapter we develop the procedure for generating **SIS** for multiple cars maneuvering on a single lane. Initially, we have considered only two vehicles in order to validate our model quickly. Then we have generalized our developed model to n cars. Though the topic is chosen as 'car following', the maneuvers may actually be different, i.e. the rear vehicles drives forward, stop behind the front vehicles, wait till the front vehicles move forward and follow the front vehicles.

In order to assure avoiding collision by the rear vehicles with the front vehicles in the **SIS**, a new model called *Driver Model* which is discussed elaborately in Section 6.2, has been developed. Due to the introduction of this new developed model, the rear vehicle starts to decelerate as soon as it perceives that the front vehicle is standing, and comes to a stop behind the front vehicle. By means of doing this experiment a doorway of opening a new research field, i.e. generating **SIS** for car maneuvers from Natural Language Descriptions as distinguished from conceptual descriptions used here, have been come to know. Section 6.4 describes this point more clearly.

The following section discusses about the problems which we faced while carrying out experiments and the solution by which we tackled those problems to attain our expected goal.

6.1 Initialization

The conceptual description for all vehicles except the agent[1] is the same as the one which we encountered already while experimenting with *single objects*. Now, we have encountered an additional set of predicates which describe the relations between

[1]means the vehicle chosen for obtaining an additional predicate

objects associated with the agent which may be either the first vehicle or the last vehicle entering into the field of view. Those are given below:

```
distance(obj_1,obj_2,d),
difference(obj_1,obj_2,similar),
relative_position(obj_1,obj_2,in_front_of)
relative_position(obj_2,obj_3,in_front_of)
```

where d is 0, 1, 2, or 3 which represents a conceptual 'distance' characterization which depends on the speed of both front and rear vehicle produced by Xtrack based on the Euclidean distance between both the vehicles. The predicate 'difference' refers to whether both vehicles drive in the same direction or in the opposite direction. The last predicate gives the information about which vehicle drives in front and which one drives behind.

Out of all additional predicates which are listed above, the one which refers the relative position has been exploited. For example, here, object 1 drives in front of object 2 which in turn drives in front of object 3. Now the query arises where to initialize the rear vehicles? First, we solved by initializing the rear vehicles from the mid point of the first segment of the initial lane as we did previously in the case of *single* moving vehicles. But it may cause collision if a vehicle is initialized and the corresponding initial position on the lane is already occupied by another vehicle. This is being noticed particularly when the number of vehicles are more than four.

Moreover, we wanted to initialize the vehicles such that the initial position will be very close to the corresponding initial position of the vehicles in the **OIS**. Since vehicle tracking in Xtrack has usually been initialized as soon as the object has been completely visible in the field of view, we have assumed to initialize all vehicles in the **SIS** from one fourth of the first segment of the initial lane. The main problem is that we only know the initial *lane* which is occupied by an object rather than its exact initial *position* within this lane. What we assumed will hold good unless we may get a predicate describing initial position of the object as

```
initial(obj_1,lane_1,5meter from traffic signal)
```

6.2 Driver Model

After solving the problem of initializing the vehicles, we have encountered another devastating problem of collision by the rear vehicles with the front vehicles while they are maneuvered based on the conceptual description using the circular motion model

with varying angular velocity which has been discussed already in Chapter 5. This is being particularly noticed when the time gap between initializing successive vehicles is comparatively small. In order to avoid this problem, a new model called '*Driver Model*' has been developed. The following few paragraphs discuss briefly the new model.

In a real traffic intersection scenery, the driver in a vehicle which basically drives towards an intersection, starts to reduce the speed of the vehicle as soon as he sees some vehicle is standing in front of a traffic signal post and waiting to get a green signal. It means that the driver will decelerate by applying the brake and stop behind the front vehicle such that the possibility of collision should not arise. Based on this practical concept, a 'Driver Model' has been developed in which the rear agent starts to decelerate his vehicle as soon as he perceives that the front vehicle is standing.

As soon as the rear agent is going to decelerate, the successive speed is calculated by the following formula until it comes to stop:

$$v(t) = v_0 + a * t \quad ,$$

where $v(t)$ refers the speed required for calculating the next successive position, v_0 refers the current speed of the vehicle. t refers to the time step between two successive poses. a represents the deceleration which is calculated in the following way:

$$a = \frac{-v_0^2}{2 * s} \quad ,$$

where s represents the current distance between current position of the rear vehicle and the position where it has to come to a stop behind the front vehicle.

Moreover, in order to reduce the gap between two vehicles while standing we have assumed that the rear vehicle will decelerate only after it crosses the *critical distance* which is the minimum distance assumed to be required for decelerating in order to avoid collision. The assumed value for the critical distance is taken as 10 meters. Similarly, we have assumed a *minimum gap* of 1 meter between two vehicles after the rear vehicle has come to a stop.

Here, the value which we have assumed for both *critical distance* and *minimum gap* between two vehicles is purely based on heuristic. In practice, these values will depend on the aggressiveness of the driver, i.e. the more aggressive driver will apply the break suddenly to come to a stop after he has driven closer to the front vehicle and he will stop just behind the front vehicle or vice-versa.

Due to uncertainties in the speed estimation by the Xtrack system, the estimated speed of the front vehicle initially oscillates around zero which is shown in fig 6.1. These oscillation are so strong that the estimated speed on the conceptual level is also

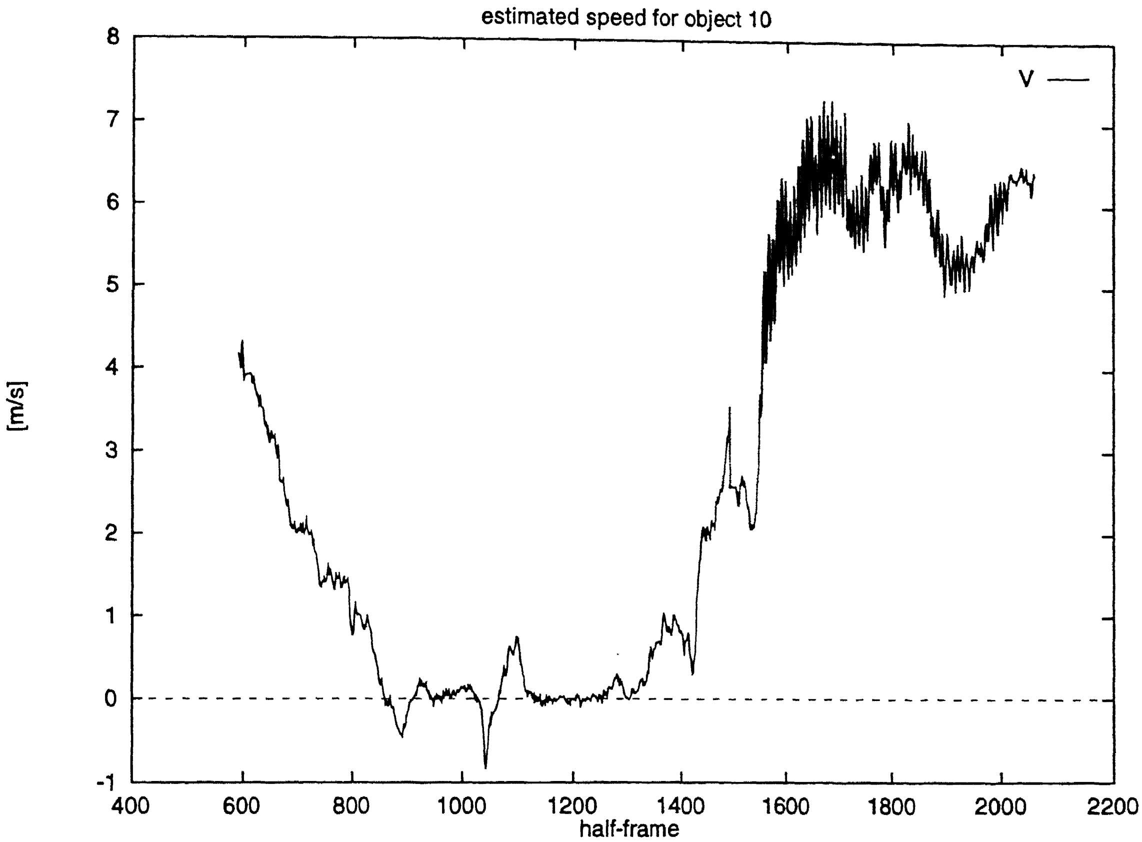


Figure 6.1: The speed for the front object estimated by the Xtrack system indicates that the speed oscillates around zero between half frame times 830 to 1100 though the object in **OIS** stands.

affected. It seems that the front vehicle has started immediately after it has come to a stop, although the front vehicle in **OIS** actually stands. This will cause the rear agent to drive forward according to the speed obtained from the conceptual description. Now we have identified another possibility of arising a collision. If both vehicles are moving and the speed of the rear vehicle is higher than the speed of the front vehicle, it may cause a collision particularly when the rear vehicle is driving closer to the front vehicle. In order to avoid collision arisen in this way also, an assumption has been made that the speed of the rear vehicle will be at most the speed of the front vehicle when vehicles are driving at close distance, for example at an assumed distance of 5 meter.

6.3 Results

We have achieved the essential maneuvers for multiple cars moving on a single lane without collision by introducing a *Driver Model*. We have tested our developed model by this way with different car following image sequences in which 2, 3, 4 and 5 car maneuver on a single lane. The results obtained by our system are discussed below. Moreover, based on experiments, the assumed default value for the 'normal' attribute of the 'speed' predicate has been taken as 25 km/h instead of 35 km/h. By this assumption, the difference in position between **SIS** and **OIS** has been reduced significantly which allows us to produce a better match between **SIS** and **OIS**.

Figure 6.2 shows the start pose of the first vehicle entering into the field of view for a 2-car and a 5-car following sequence. The question of where to initialize the vehicles is resolved by initializing at one fourth of the first segment of the initial lane. By doing so the initial lag between **SIS** and **OIS** is more or less eliminated. This is being noticed for other vehicles as well as for other car following image sequences.

The top panel in Figure 6.3 shows the pose of the front and rear vehicle at half frame time point 1153. It indicates that though the rear vehicle perceives that the front vehicle is standing from starting half frame point itself, it starts to decelerate only after it has reached the region of *critical distance*. By imposing this condition, the rear vehicle comes to a stop just behind the front vehicle as in the **OIS** as shown in the bottom panel of Figure 6.3. The rear vehicle in the **OIS** is occluded by a traffic signal post.

Figure 6.4 shows the pose of all vehicles at half frame time point 1023 for a 3-car following image sequence. From this half frame point to half frame point 1142, the speed of the rear objects has been taken to be the speed of their corresponding front vehicle. Because, as we mentioned in the previous section, between these time frame points, the estimated speed of rear objects can be higher than the speed of their corresponding predecessor vehicle. In this case, the **SIS** vehicle would collide with its predecessor. This is being noticed while experimenting with 4 and 5 objects.

Figure 6.5 shows the pose of all vehicles which are all waiting at an intersection, for a 3-car and 5-car following image sequence. Notice, no cars are colliding with their corresponding front vehicle. This is what we essentially expected.

Figure 6.6 shows the pose of all vehicles which are driving forward after leaving the intersection, for a 4-car and 5-car following sequence. This indicates that the rear vehicles follow their front vehicles exactly as indicated by the input obtained from the conceptual description.

6.4 Discussion

As we mentioned before, the possibility of opening a new field has come, i.e generating **SIS** for car maneuvers from Natural Language Descriptions. The experiment which we did can be done with the input from Natural Language Text as 'object 1 drives forward on lane 1 at a normal speed, stops behind object 2 and follows object 2'. The first phase can be done as a normal maneuver by initializing the vehicle on a corresponding lane with the mentioned speed. The second phase can be achieved by the Driver Model. The last one can be done as a special maneuver in which the speed of the rear vehicle will be considered as a function of the speed of the first vehicle.

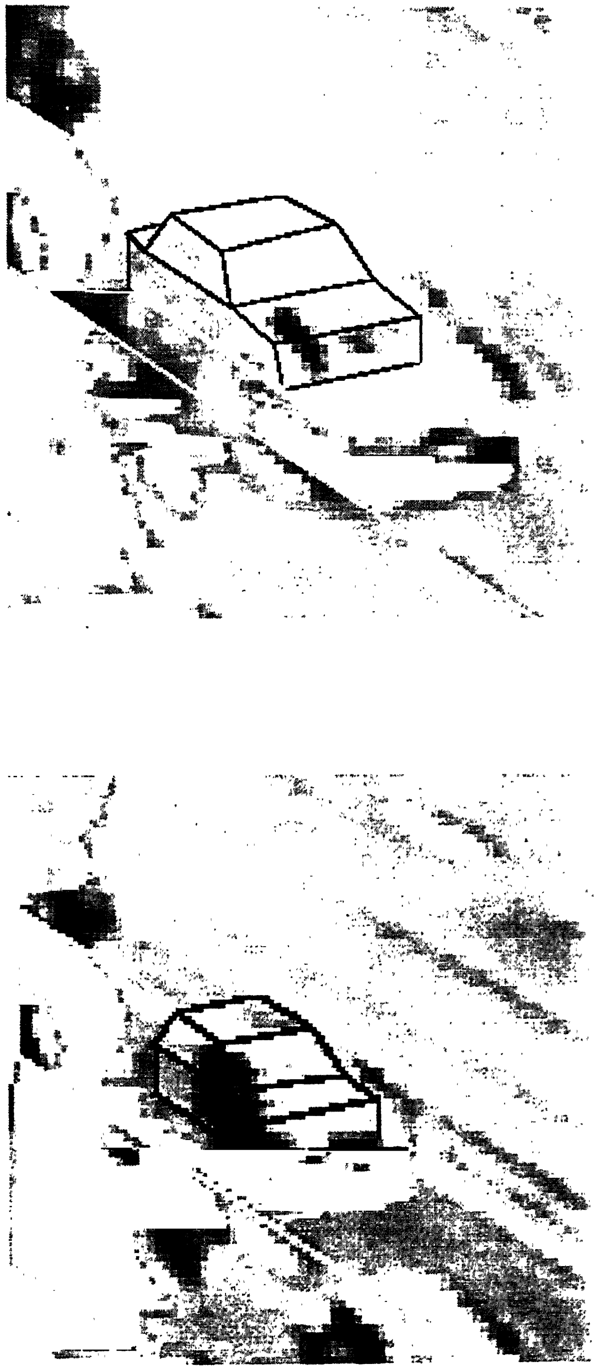


Figure 6.2: The poses of the first vehicle, object 3, at starting time of 350 for 2 cars in stau02 (top) and, the object 3, for 5 cars in stau10 (bottom) car following image sequences indicate that the initial lag between **SIS** and **OIS** is mostly eliminated by initializing the vehicle at one fourth of the first segment.

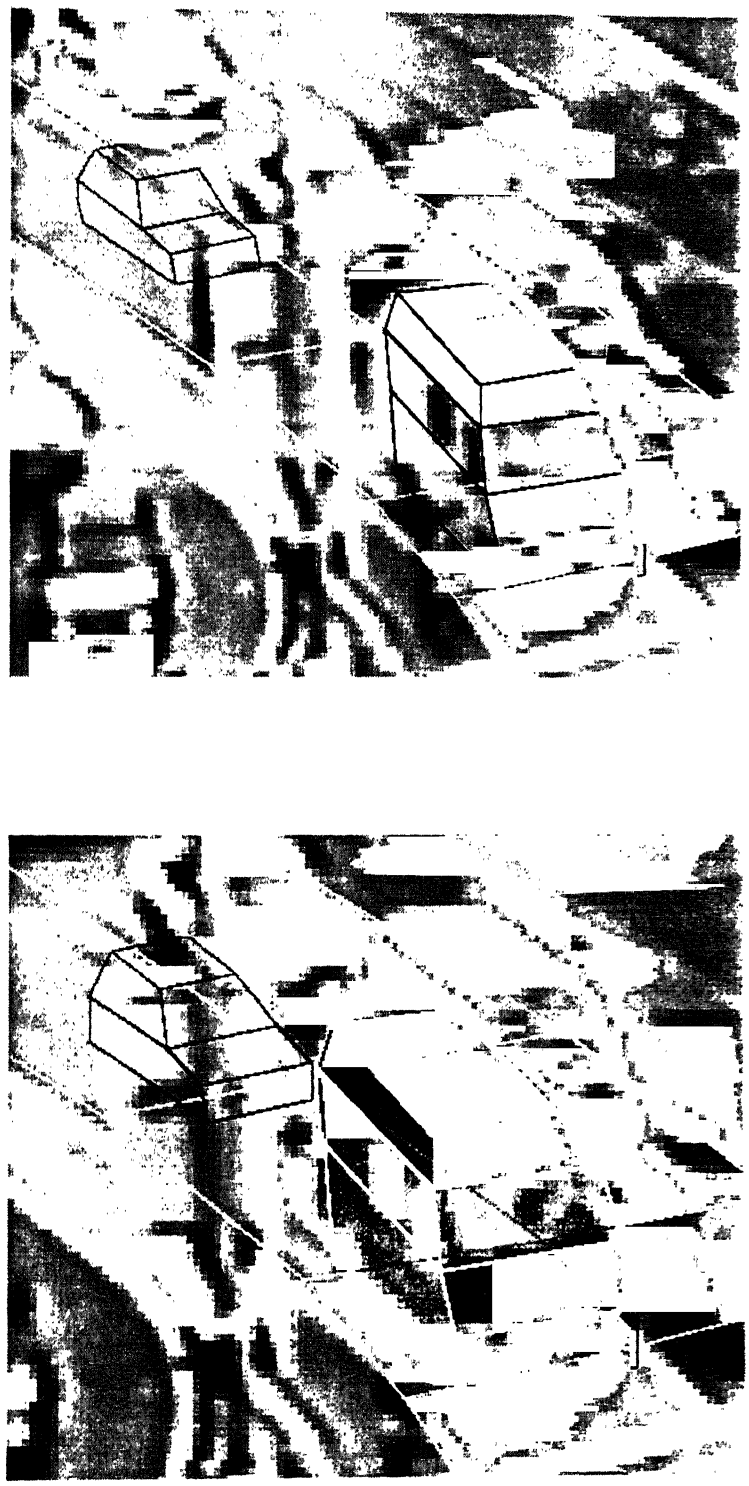


Figure 6.3: The pose of object 2 & 6 for 2 cars in stau02 following sequence from which the rear vehicle starts to decelerate (top) and the pose after it has come to a stop behind the front vehicle(bottom).

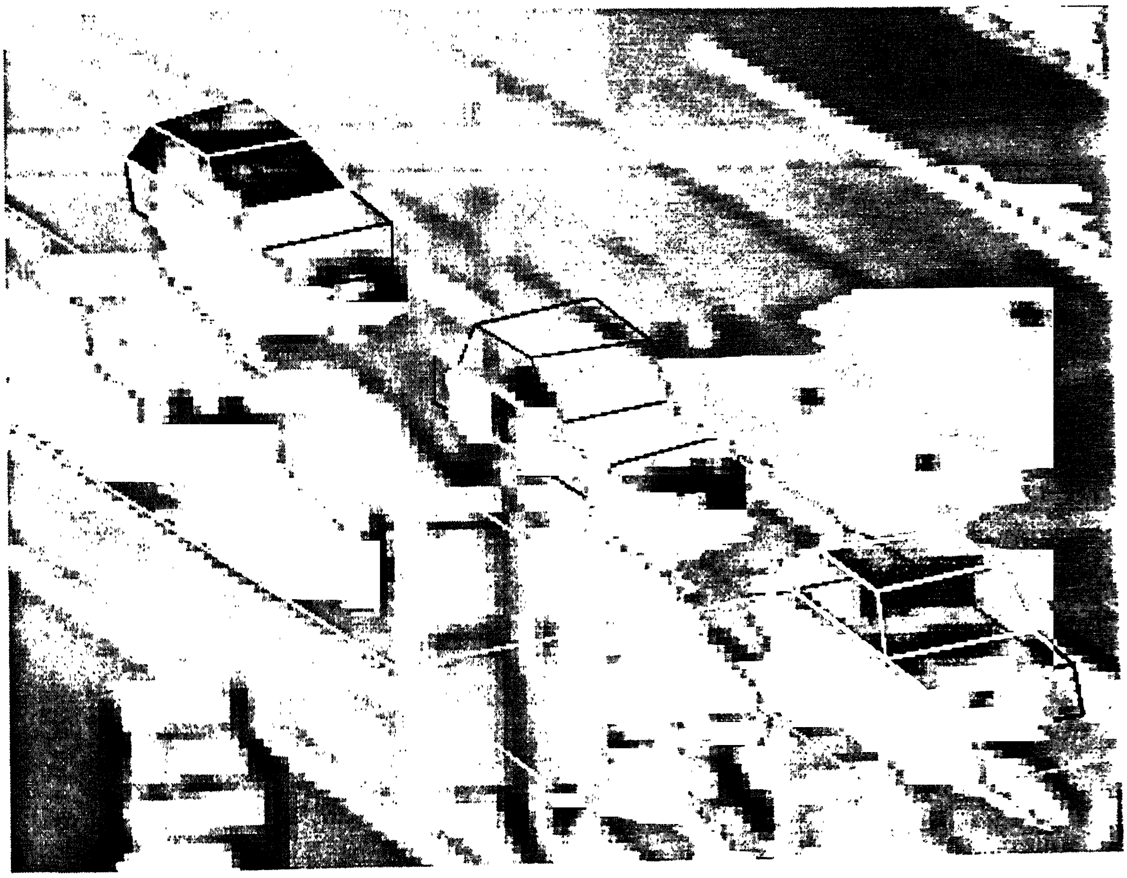


Figure 6.4: The pose of all vehicles, namely object 6, 9 & 10, for 3 cars in the 'stau09' following sequence at half frame time point 1023. From this time the rear vehicles follow their front vehicles until half frame time point 1142.

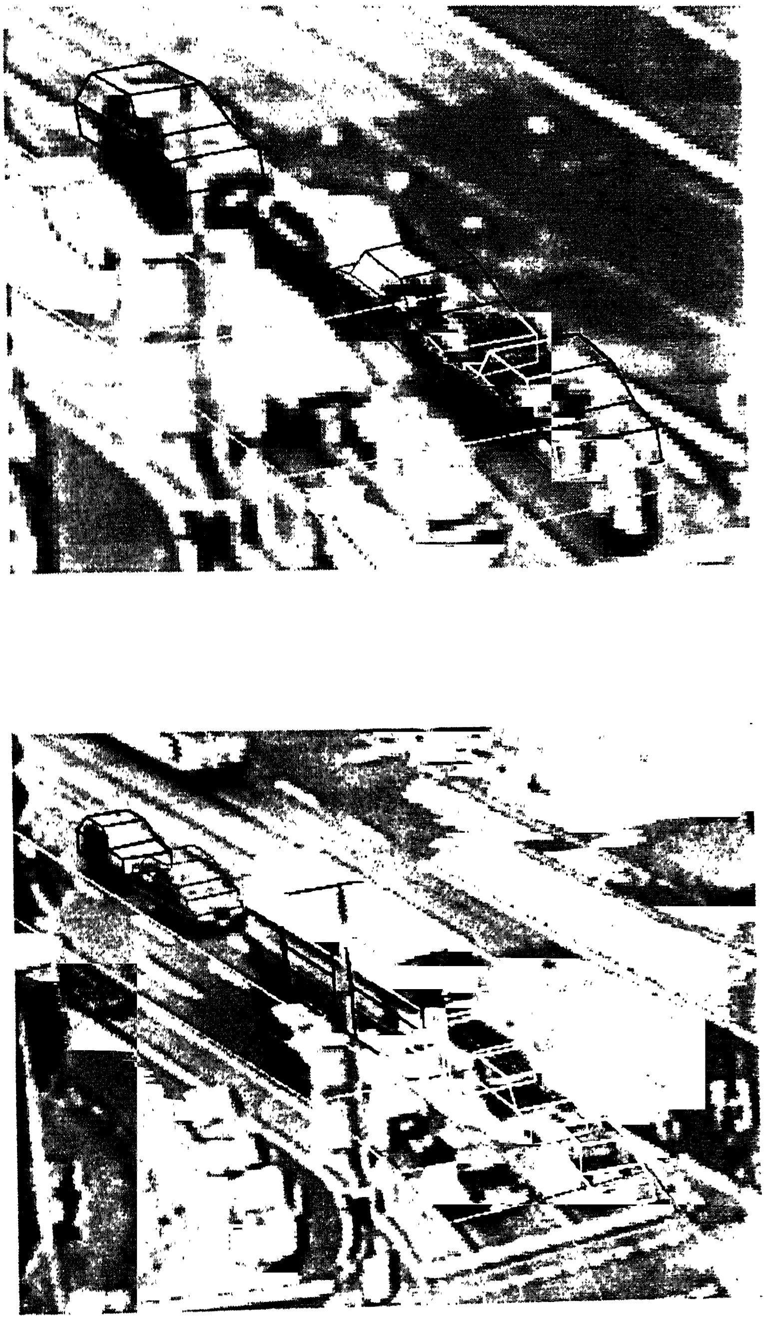


Figure 6.5: The estimated pose of all vehicles - object 6, 9 & 10 – for 3 cars in 'stau09' (top) and object 3, 7, 12, 16 & 31 for 5 cars in 'stau10' (bottom) image sequence indicate that all vehicles are waiting at an intersection.

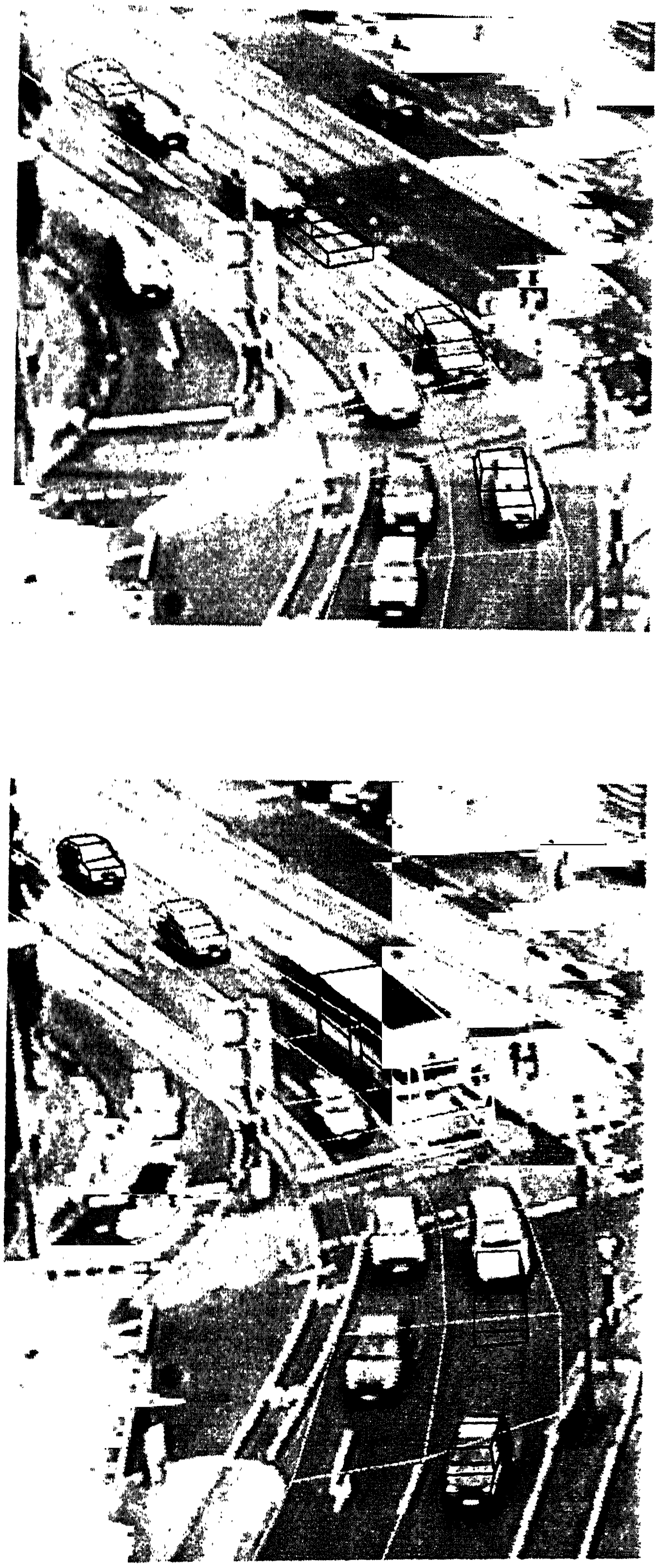


Figure 6.6: The pose of all vehicles, namely object 14, 19, 27 and 34, at half frame time point 3750 for 4 cars in 'staul2' (top) and at 2950 for a 5 cars in 'staul0' following sequence (bottom) indicate that all vehicles are driving forward after getting a green signal.

Chapter 7

Lane Changing

In this chapter, the procedure for generating SIS for the most complicated task of overtaking is reported. The essential maneuver for overtaking could be that changing the position from one lane to another parallel lane, overtaking the front vehicle, then once again shifting to the previous lane. Among the essential maneuvers, the key maneuver will be shifting from one to another lane. This is the one which we have accomplished here. The following section describes how the vehicle is maneuvered while lane changing.

7.1 Maneuvering

The *Circular Motion Model with varying angular velocity* is used which has already been discussed in section 5.3 while the vehicle is maneuvering in one lane, i.e from its start position to the position where it is going for lane changing maneuver, and after the lane changing. We have introduced a new motion model to maneuver the vehicle in a practical manner while lane changing.

The new motion model is nothing but a combination of two circular motion models with varying angular velocity. The first circular motion model will be constructed with the data of the current position (P1), i.e. from where it will be going for lane changing, current orientation and the target point (P2) which will be the center point of the line which connects the current position and the mid point of the destination lane. By this motion model, the vehicle will be maneuvered till the halfway between starting and ending of the lane changing maneuver.

Similarly, the second circular motion model which is used to maneuver the vehicle from halfway to end of the lane changing, will be constructed with the data of the current pose which comprise of the position (P2) and the orientation and the target

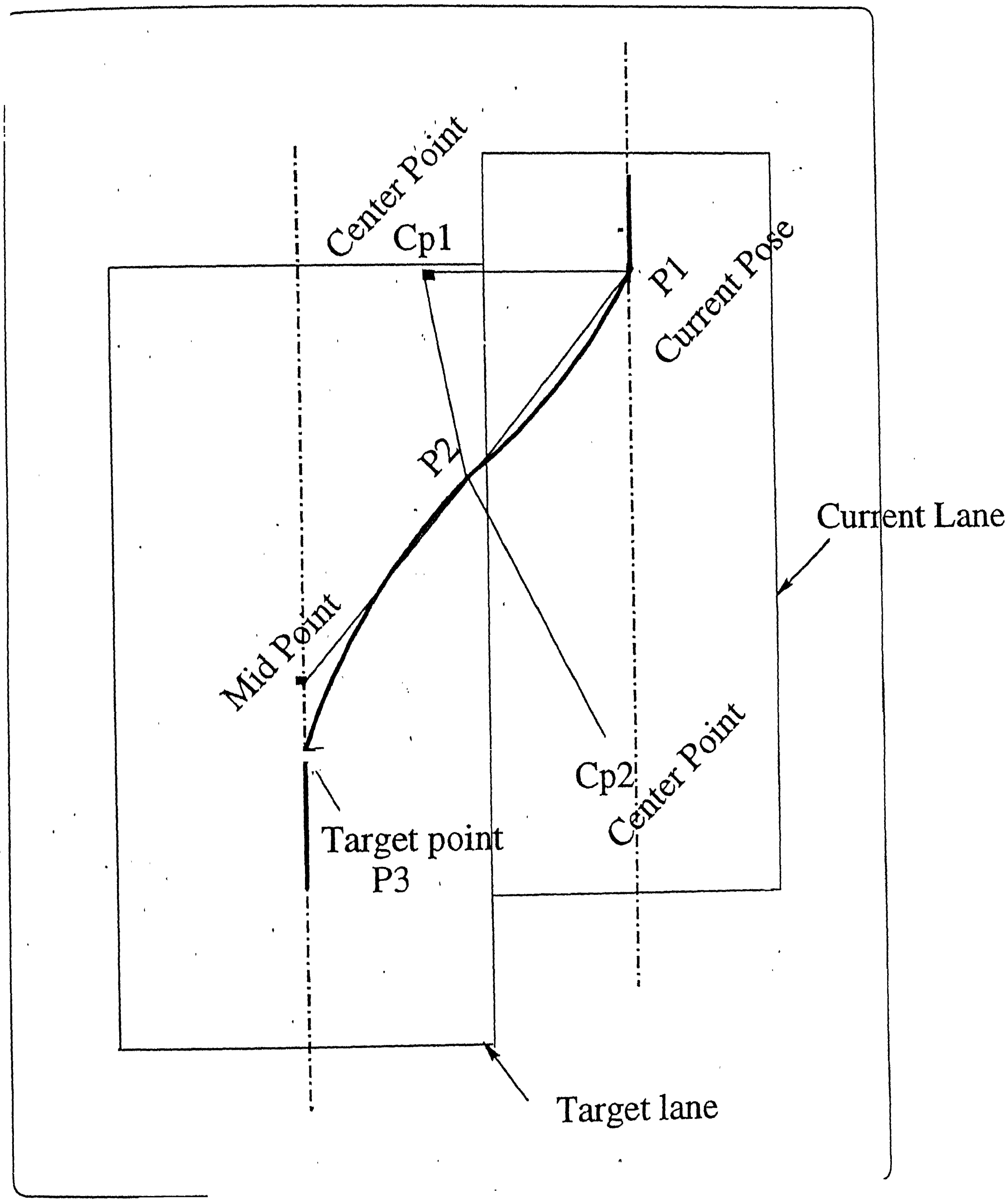


Figure 7.1: Construction of the first and second circular motion model in order to achieve the lane changing maneuver.

orientation which is the orientation of the target lane. Fig 7.1 shows how the first and the second circular motion model have been constructed.

7.2 Results

Because of unavailability of having an **OIS** with many examples of a lane changing maneuver, we have tested our improved system with only one example. The interesting results are discussed below.

Unless the starting point of the vehicle is known explicitly from conceptual descriptions, the maneuvering in the **SIS** could be initialized by giving its geometric starting point - obtained by a motion based segmentation step in the Xtrack System — interactively. By doing so, a full stop has been put to the query regarding initialization which arose while experimenting with previous tasks. Thus the vehicle position in **SIS** and **OIS** exactly matches while initializing and is shown in Figure 7.2.

Figure 7.3 shows the trajectory and the pose of the vehicle at 50 successive half frame time points. Notice the smooth trajectory which has been attained while the vehicle is in a lane changing maneuver by introducing a newly constructed motion model.

The top panel in Figure 7.4 shows the pose of the vehicle at half frame time point 3797. It indicates that the object in **SIS** starts to change its lane from a different segment as compared to the **OIS**. Similarly, the bottom panel shows the pose of the object at half frame time 3885 which indicates that the object in **SIS** attains its target pose soon after the lane changing maneuver while in **OIS** it is still in the lane changing maneuver.

7.3 Discussion

Though by testing with only one example, the essential maneuver of lane changing has been captured. Our system could be validated further by testing with many more examples of lane changing maneuvers. As we mentioned before, lane changing will be the key maneuver of *overtaking* which now could be easily accomplished by extending the developed system.

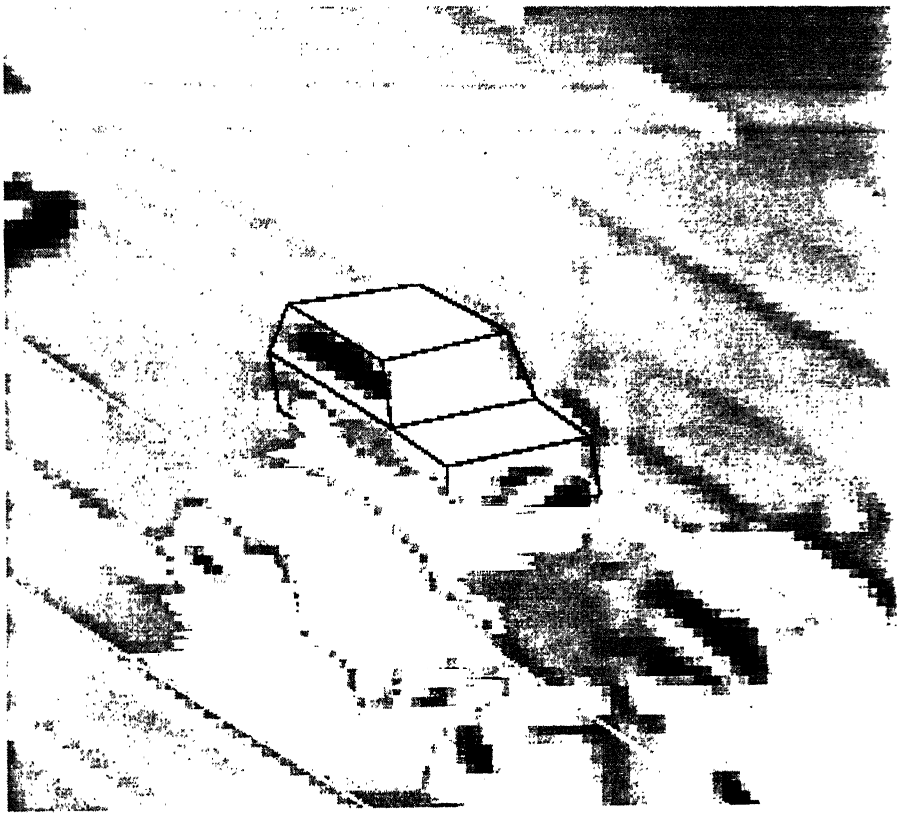


Figure 7.2: The initial pose of object 29 in 'stau08' sequence indicates that exact match between **SIS** and **OIS** could be obtained by interactively giving its starting point as obtained by the Xtrack System.

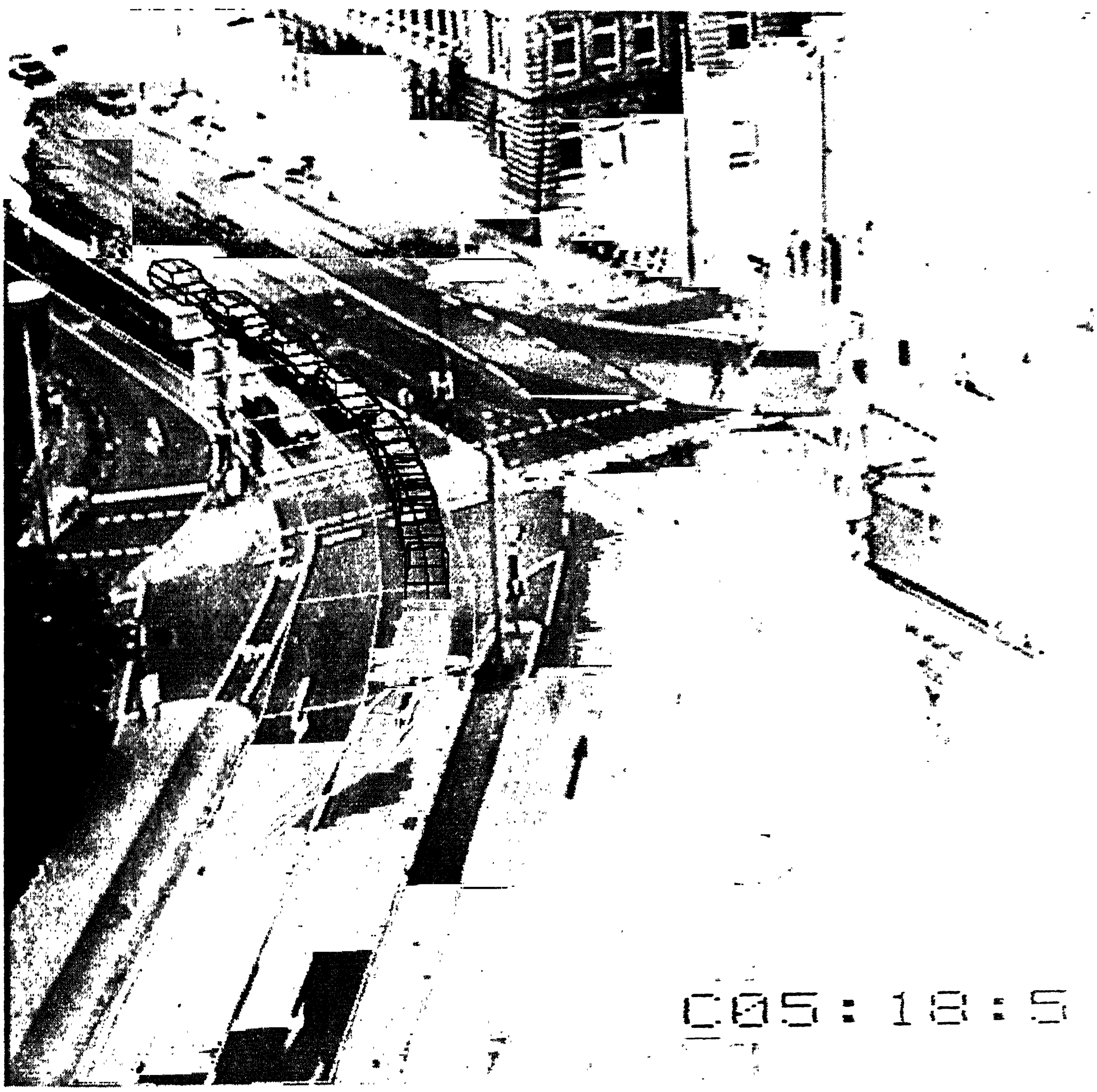


Figure 7.3: The trajectory of object 29 indicates that the smooth trajectory could be achieved while lane changing by introducing a new motion model which comprises of two circular motion models with varying ω.

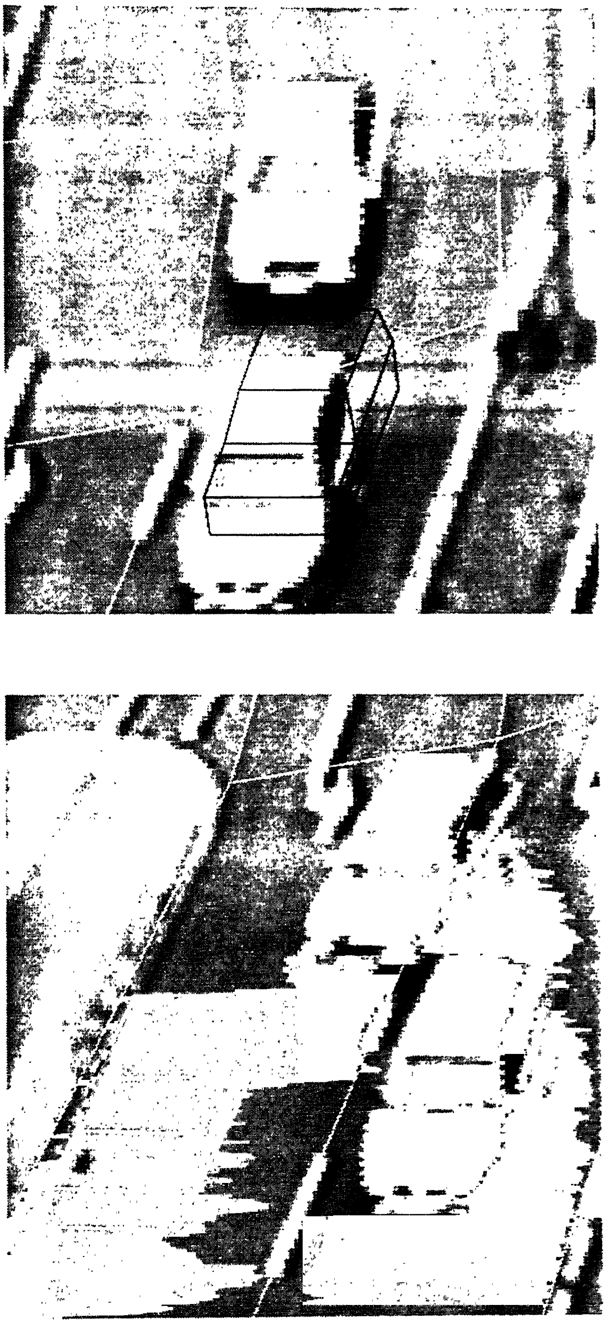


Figure 7.4: The pose of object 29 at half frame time 3797. From this position the object will be in the maneuver of lane changing (top). The pose of object 29 at half frame time 3885 (bottom). The target pose of the lane changing maneuver in **SIS** is attained soon in comparison with the **OIS**.

Chapter 8

Summary and Future Work

8.1 Summary

The polyhedral vehicle model has been projected onto the background image plane. After reading a geometric lane structure of the intersection scene and a list of conceptual primitives, the first basic maneuver of a single car driving on a straight lane has been implemented. Subsequently, maneuvering the vehicle on a curved lane has been done with the help of a newly developed model named *Circular Motion Model with Varying Angular Velocity* (Section 5.3). Next, by introducing a *Driver Model* (Section 6.2), n cars following one another has been achieved in a more practical manner. Finally, lane changing which would be the key maneuver for overtaking has been done by constructing a new motion model which is nothing but a combination of two circular motion models with varying angular velocity.

The resulting **SIS** have been compared to the **OIS** and potential reasons for the differences have been discussed. In all examples investigated here, the basic maneuvers occurring in the **OIS** have been captured in the **SIS**. Differences between **OIS** and **SIS** which are only of geometric nature, have not been considered as significant, since it is obvious that some quantitative information is lost during the transition between geometric estimation results by the Xtrack System to conceptual descriptions.

8.2 Improvements

The current developed system could be improved by the following suggestions.

8.2.1 Speed Model - Defuzzification

As we mentioned in Section 4.4, The predicate with the highest degree of validity has been chosen though several predicates may hold at the same half frame time point. By using more than one predicates as shown in Fig 8.1, intermediate values can be determined leading smoother speeds. This could be also easier if the fuzzy function is invertible - eg. if it is triangular.

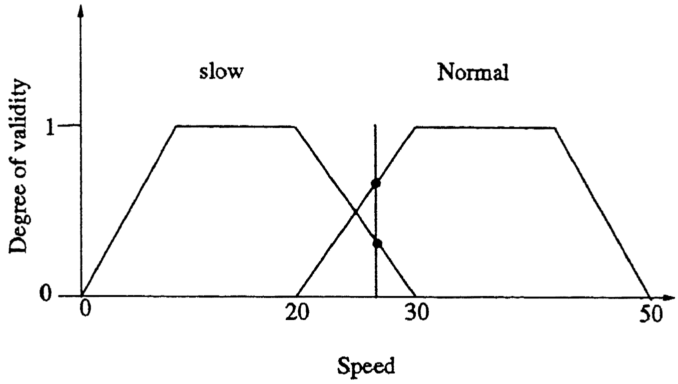


Figure 8.1: smoothening speed over time interval by using more than one predicates at a particular time frame point.

8.2.2 Speed Smoothing

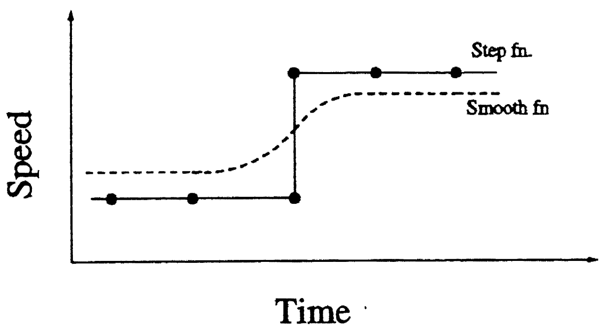


Figure 8.2: Smoothing the speed over time interval by using smooth function instead of step function.

At same points the speed may suddenly transition from one fuzzy zone to another. Currently this is being implemented as a step function (Fig 8.2). By spanning several adjacent frame steps during reconstruction, a smoother speed profile may be obtained.

8.2.3 Improving the conceptual description

The conceptual description which is used in this work represent an abstraction for the concepts of speed and distance which is also OIS distance. By introducing a concept of 'sensed distance', the SIS vehicle position w.r.t other vehicles in OIS can be presented as fuzzy model. This is one of the improvement activities that could be pursued. This would avoid situation in the current model where the SIS vehicle collides with or overruns an OIS vehicle.

8.2.4 Spatial landmarks in conceptual description

Conceptual description can have richer spatial attributes. Many landmarks such as "stopping line", "intersection", "traffic island" etc. can be used to generate abstract spatial description such as

```
position(at,stopping_line)
position(near,traffic_island)
```

These can be used to generate SIS for more precise behavior.

8.3 Future Work

The future work could be the following:

- generating **SIS** for other car maneuvers which arises possibly in real life traffic situations
- the *potential field* which is the most popular local heuristic method could be used for maneuvering the vehicle along with the inputs from conceptual descriptions.
- generating **SIS** from Natural Language Text (Section 6.4).
- once we accomplish the second one, **SIS** could be generated orally

Appendix A

DAAD Programm

DAAD, which means 'German Aca ... eme which allows foreign students for short ti ... e in research activities. This is the first tim ... are pursuing M.Tech at various IITs for doing ... ving universities.

- University of Karlsruhe, Karlsruhe
- University of Stuttgart, Stuttgart
- University of Berlin, Berlin
- University of Darmstadt, Darmstadt
- Uinverstity of Achen, Achen

The time duration for this programme is 9 months. Out of 9 months, the first month will go for German courses, followed by 6 months for Thesis work and the last two months will go for Industrial Training in any one of the top Company in Germany.

Bibliography

[Buxton & Gong 95] H. Buxton and S. Gong: *Visual Surveillance in a Dynamic and Uncertain World.* Artificial Intelligence **78**:1–2 (1995) 431–459.

[Gerber & Nagel 96b] R. Gerber and H.–H. Nagel: *Knowledge Representation for the Generation of Quantified Natural Language Descriptions of Vehicle Traffic in Image Sequences.* Proc. IEEE International Conference on Image Processing (ICIP'96), Lausanne/CH, 16-19 September 1996, Vol. II, pp. 805–808.

[Gerber & Nagel 98] R. Gerber and H.–H. Nagel: *(Mis?)-using DRT for Generation of Natural Language Text From Image Sequences.* Proc. Fifth European Conference on Computer Vision (ECCV'98), Freiburg/Germany, 2-6 June 1998; H. Burkhardt and B. Neumann (Eds.), Lecture Notes In Computer Science **1407** (Vol. II), Springer-Verlag Berlin, Heidelberg, New York 1998, pp. 255–270.

[Gupta et al. 98] S. Gupta, P. chakraborty, and A. Mukherjee: *Microscopic Simulation of Vehicular Traffic on Congested Roads.* Proc. ISIRS'98, M. Vidyasagar (Ed.), Bangalore, 10-13 January 1998;

[Haag et al. 97b] M. Haag, W. Theilmann, K.H. Schäfer, and H.–H. Nagel: *Integration of Image Sequence Evaluation and Fuzzy Metric Temporal Logic Programming.* KI-97: Advances in Artificial Intelligence, Proc. 21st Annual German Conference on Artificial Intelligence, Freiburg, Germany, 9-12 September 1997, G. Brewka, C. Habel, B. Nebel (Eds.) Lecture Notes in Artificial Intelligence **1303**, Springer-Verlag Berlin, Heidelberg, New York 1997, pp. 301–312.

[Haag & Nagel 98a] M. Haag and H.–H. Nagel: *Incremental Recognition of Traffic Situations from Video Image Sequences.* Proc. ICCV-98 Workshop on Conceptual Descriptions of Images (CDI-98), 2 January 1998, Bombay/India, H. Buxton and A. Mukherjee (Eds.), pp. 1–20.

[Haag & Nagel 98b] M. Haag and H.–H. Nagel: *Beginning a Transition from a Local to a More Global Point of View in Model-Based Vehicle Tracking.*

Proc. Fifth European Conference on Computer Vision (ECCV'98), Freiburg/Germany, 2-6 June 1998; H. Burkhardt and B. Neumann (Eds.), Lecture Notes in Computer Science **1406** (Vol. I), Springer-Verlag Berlin, Heidelberg, New York 1998, pp. 812–827.

[Horn 86] B.K.P. Horn: *Robot Vision.* The MIT Press, Cambridge/MA and London/UK 1986.

[Kamp & Reyle 93] H. Kamp and U. Reyle: *From Discourse to Logic.* Kluwer Academic Publishers, Dordrecht, Boston/MA, London 1993.

[Koller 92] D. Koller: *Detektion, Verfolgung und Klassifikation bewegter Objekte in monokularen Bildfolgen am Beispiel von Strassenverkehrsszenen.* Dissertation, Fakultät für Informatik der Universität Karlsruhe (TH), Juni 1992; erschienen in Dissertationen zur Künstlichen Intelligenz (DISKI) **13**, infix-Verlag, St. Augustin 1992.

[Koller et al. 93] D. Koller, K. Daniilidis, and H.-H. Nagel: *Model-Based Object Tracking in Monocular Image Sequences of Road Scenes.* International Journal of Computer Vision **10**:3 (1993) 257–291.

[Kollnig & Nagel 96] H. Kollnig and H.-H. Nagel: *Matching Object Models to Segments from an Optical Flow Field.* Proc. Fourth European Conference on Computer Vision 1996 (ECCV'96), 15-18 April 1996, Cambridge/UK; B. Buxton and R. Cipolla (Eds.), Lecture Notes in Computer Science **1065** (Vol. II), Springer-Verlag Berlin, Heidelberg 1996, pp. 388–399.

[Kollnig & Nagel 97] H. Kollnig and H.-H. Nagel: *3D Pose Estimation by Directly Matching Polyhedral Models to Gray Value Gradients.* International Journal of Computer Vision **23**:3 (1997) 283–302.

[Howrath & Buxton 98] R. Howrath and H.Buxton: *Conceptual description from monitoring and watching image sequences.* Proceedings on Conceptual Description of images '98.(1998).

[Krüger 91] W. Krüger: *Begriffsgraphen zur Situationsmodellierung in der Bildfolgenauswertung.* Dissertation, Fakultät für Informatik der Universität Karlsruhe, Januar 1991. Also available as: Informatik-Fachberichte **311**, Springer-Verlag, Berlin, Heidelberg, New York/NY 1992.

[Mukerjee et al. 98] A. Mukherjee, K. Gupta, S. Nautiyal and M. P. Singh: *Conceptual Description of Visual Scenes from Linguistic Models.* Proc. ICCV-98 Workshop on Conceptual Descriptions of Images (CDI-98), 2 January 1998, Bombay/India, H. Buxton and A. Mukherjee (Eds.).

[Nagel 88] H.-H. Nagel: *From Image Sequences Towards Conceptual Descriptions.* Image and Vision Computing **6**:2 (1988) 59–74.

[Nagel 91] H.–H. Nagel: *The Representation of Situations and their Recognition from Image Sequences.* Proc. 8e Congres Reconnaissance des Formes et Intelligence Artificielle, Lyon-Villeurbanne, 25-29 Novembre 1991, pp. 1221–1229.

[Nagel et al. 95] H.–H. Nagel, H. Kollnig, M. Haag, and H. Damm: *The Association of Situation Graphs with Temporal Variations in Image Sequences.* In: Working Notes AAAI-95 Fall Symposium Series 'Computational Models for Integrating Language and Vision', 10-12 November 1995, pp. 1–8, Cambridge/MA, USA.

[Nagel & Haag 98] H.–H. Nagel and M. Haag: *Bias-Corrected Optical Flow Estimation.* Proc. Sixth International Conference on Computer Vision 1998 (ICCV'98), 4-7 January 1998, Bombay/India, Narosa Publishing House New Delhi a.o. 1998, pp. 1006–1011.

[Nagel et al. 98] H.–H. Nagel, Th. Schwarz, H. Leuck, M. Haag: *Tracking Turning Trucks with Trailers.* IEEE Workshop on Visual Surveillance, January 2, 1998, Bombay/India.

[Otte & Nagel 95] M. Otte and H. –H. Nagel: *Estimation of Optical Flow Based on Higher-Order Spatiotemporal Derivatives in Interlaced and Non-interlaced Image Sequences.* Artificial Intelligence **78**:1–2 (1995) 5–43.

[Schäfer 97] K. Schäfer: *Fuzzy Spatio-Temporal Logic Programming.* In C. Brzoska (Ed.): Proc. of the Workshop #27 "Programming in Temporal and Non-Classical Logics", International Joint Conference on Artificial Intelligence IJCAI-97, 23-28 August 1997, Nagoya, Japan.